SHARP MZ—731

个人计算机使用手册

三来电子公司

1988年

目　　录

第一章　MZ—731个人计算机系列领域

第一节　MZ—700系列机特点……………………………………1
第二节　说明书的使用……………………………………………1
第三节　步入计算机领域…………………………………………2

第二章　BASIC语言

第一节　BASIC程序设计入门……………………………………5
第二节　BASIC语言程序设计……………………………………10
第三节　常用的BASIC命令和语句………………………………15
第四节　内部函数…………………………………………………55
第五节　字串函数…………………………………………………59
第六节　彩色显示语句……………………………………………63
第七节　彩色绘图打印命令………………………………………65
第八节　机器语言程序的控制语句………………………………78
第九节　输入／输出（I／O）语句………………………………81
第十节　其他语句…………………………………………………82
第十一节　监控程序的作用………………………………………85

第三章　MZ—700的操作

第一节　MZ—700系列个人计算机的外貌…………………………89
第二节　连接显示器………………………………………………90
第三节　数据录音机………………………………………………92
第四节　彩色绘图打印机…………………………………………94
第五节　键盘操作…………………………………………………98

第四章　硬件

第一节　MZ—700系统图…………………………………………103
第二节　内存配置…………………………………………………104
第三节　I／O（$E000—$E008）输入／输出口的内存分配………109

第四节　颜色V—RAM的信号系统……111
第五节　MZ—700线路图……113

第五章　监控命令和子程序

第一节　监控命令……124
第二节　监控命令功能的应用……124
第三节　监控程序的子程序……128

附　　录

A·1　代码表……132
A·2　MZ—700系统计算机技术指导……136
A·3　BASIC错误信息表……138
A·4　Z80A指令系统……140
A·5　监控程序汇编列表……144
A·6　彩色绘图打印机的控制代码……145
A·7　注意有关操作……148

附录A8

A8·1　如何编写程序……154
第一节　编写程序的步骤……154
第二节　程序优化……157
A8·2　几种简单计算方法的BASIC程序……158
第一节　用牛顿法解代数方程……159
第二节　配直线回归方程……160
第三节　打印衰减震荡函数$e^{-0.12x}$·SInx的曲线……161
第四节　用辛普生公式求数值积分……161
第五节　用高斯消去法解线性方程组……162

指令表　括弧（　）为简化格式

A

ABS 55
ASC 61
ATN 56
AUTO (A.) 16
AXIS (AX.) 75

B

BYE (B.) 20

C

CHR$ 61
CIRCLE (CI.) 76
CLOSE (CLO.) 52
CLR 43
COLOR (COL.) 63
CONSOLE (CONS.) 85
CONT (C.) 19
COS 56
CURSOR (CU.) 45

D

DEF FN 40
DEF KEY 41
DELETE (D.) 17
DIM 39

E

END (E.) 43
EXP 56

F

FOR~NEXT (F.~N.) 30

G

GET 27
GOSUB
 ~RETURN (GOS.~RET.) 33
GOTO (G.) 32
GPRINT (GP.) 74

H

HSET (H.) 74

I

IF ERL 82
IF ERN 83
IF~GOSUB (IF~GOS.) 37
IF~GOTO (IF~G.) 36
IF~THEN (IF~TH.) 34
INP 81
INPUT (I.) 26
INPUT/T (I./T) 52
INT 55

J

K

KEY LIST (K. L.) 20

L

LEFT$ 60
LEN 59
LET 20
LIMIT (LIM.) 78
LINE 70
LIST (L.) 17
LIST/P (L./P) 17
LN 56
LOAD (LO.) 15
LOG 56

M

MERGE (ME.) 18
MID$ 60
MODE GR (M. GR) 67
MODE TL (M. TL) 67
MODE TN (M. TN) 66
MODE TS (M. TS) 67
MOVE 72
MUSIC (MU.) 50

N

NEW 18

O

ON ERROR
  GOTO .... (ON ERR. G.) 82
ON~GOSUB .... (ON~GOS.) 38
ON~GOTO .... (ON~G.) 37
OUT 81

P

PAGE 68
PAI 56
PCOLOR .... (PC.) 67
PEEK 80
PHOME .... (PH.) 73
PLOT OFF .... (PL. OFF) 85
PLOT ON .... (PL. ON) 84
POKE 79
PRINT .... (?) 21
PRINT USING .... (?USI.) 22
PRINT/P .... (?/P) 69
PRINT/T .... (?/T) 52
PRINT [α, β] .... (? [α, β]) 64

Q

R

RAD 56
READ~DATA .... (REA. ~DA.) 28
REM 41
RENUM .... (REN.) 18
RESET 47
RESTORE .... (RES.) 29
RESUME .... (RESU.) 83
RIGHT$ 60
RLINE .... (RL.) 71
RMOVE .... (RM.) 73
RND 57
ROPEN .... (RO.) 52
RUN .... (R.) 19

S

SAVE .... (SA.) 15
SET 47
SGN 55
SIN 55
SIZE 84
SKIP 68
SPC 46
SQR 56
STOP .... (S.) 42
STR$ 62

T

TAB 46
TAN 56
TEMPO .... (TEM.) 51
TEST .... (TE.) 68
TI$ 44

U

USR .... (U.) 80

V

VAL 62
VERIFY .... (V.) 16

W

WOPEN .... (W.) 52

X

Y

Z

N

NEW 18

O

ON ERROR
  GOTO .... (ON ERR. G.) 82
ON~GOSUB .... (ON~GOS.) 38
ON~GOTO .... (ON~G.) 37
OUT 81

P

PAGE 68
PAI 56
PCOLOR .... (PC.) 67
PEEK 80
PHOME .... (PH.) 73
PLOT OFF .... (PL. OFF) 85
PLOT ON .... (PL. ON) 84
POKE 79
PRINT .... (?) 21
PRINT USING .... (?USI.) 22
PRINT/P .... (?/P) 69
PRINT/T .... (?/T) 52
PRINT [α, β] .... (? [α, β]) 64

Q

R

RAD 56
READ~DATA .... (REA. ~DA.) 28
REM 41
RENUM .... (REN.) 18
RESET 47
RESTORE .... (RES.) 29
RESUME .... (RESU.) 83
RIGHT$ 60
RLINE .... (RL.) 71
RMOVE .... (RM.) 73
RND 57
ROPEN .... (RO.) 52
RUN .... (R.) 19

S

SAVE .... (SA.) 15
SET 47
SGN 55
SIN 55
SIZE 84
SKIP 68
SPC 46
SQR 56
STOP .... (S.) 42
STR$ 62

T

TAB 46
TAN 56
TEMPO .... (TEM.) 51
TEST .... (TE.) 68
TI$ 44

U

USR .... (U.) 80

V

VAL 62
VERIFY .... (V.) 16

W

WOPEN .... (W.) 52

X

Y

Z

第一章　MZ—731个人计算机系列领域

第一节　MZ—700系列机特点

在过去的数十年内，计算机出现了戏剧性变化，从当初的价值昂贵，重达几十吨的庞然大物演变成今天价值低廉、轻巧玲珑的设备。过去，计算机只能为少数专家所用，可是今天她已成为大多数人不可缺少的工具，夏普MZ—700系列机就是这种机器的代表者。

人类通过文字和语言交换思想。

然而夏普MZ—700系列机与人的沟通则是通过普通电视机和专用打印机　同时不论是黑白电视机，还是彩色电视机均可使用。但是如果用户愿意多花些钱的话，可以使用专用显示器，这样你将获得更加美观的屏幕显示了

为了录制程序，用户可以把磁带录音机与MZ—700系列机联接起来，通过录音机控制计算机操作。如果需要打印程序或计算结果的话，还可方便地使用MZ—700系列的轻巧而造型精美的四色绘图打印机。

注意：在本说明书中说的MZ—700指的是MZ—700系列机中的任一型号如MZ—731，MZ—721，MZ—711等。

第二节　说明书的使用

在开始练习程序设计之前，不必读完这本说明书，只要你读了第三章“MZ—700的操作”MZ—700与电视机的联接及使用键盘的过程说明，你就可以在MZ—700机上作练习了。

倘若你是第一次使用MZ—700，最好按顺序地阅读第一和第二章。开始，你可能会觉得BASIC程序设计语言中的各种指令和语句难于理解，然而即使你还未弄懂这些解释的意思，不妨按书中的例子全送入计算机，只要你这样做了，你将会渐渐地理解整个BASIC的基本概念。

你暂时可不必阅读第二章第8节机器语言程序的内容。然而，当你完全理解了BASIC语言程序设计后，再去读它，你会觉得非常有用，且也会使你更加熟悉计算机内部运算。

如果你使用过MZ—80K型号电脑，你将会发现MZ—700使用的BASIC的指令和语句与SP—5025系列机是大同小异的。因此，使用MZ—700跟使用MZ—80K是十分相似的。唯一不同点是MZ—700多了颜色语句（使用电视机，彩色绘图打印机）但只要你

详细地阅读了第二章第六节“彩色显示语句”和第二章第七节“彩色绘图打印机命令”后，就很容易掌握它了，这样你就会很快理解扩展BASIC的功能了。

本说明书，第三章是“MZ—700的操作”第四章是论述“硬件”第五章是描述“监视器命令和子程序以及其它有用信息。”

你应该进一步学习MZ—700的所有内容，这本说明书对你来说是很有帮助的。

第三节　步入计算机领域

1·3、1　什么是BASIC语言？

人类使用语言交流思想意识，计算机与人之间也是通过“专门语言”来对话的，而计算机BASIC语言便是其中之一。(BASIC是Beginner'S ALL－purpose symbolic Instruction code”的缩写，意为初学者通用符号指令码)

正如人与人之间为了交流联络，使用英语、法语、德语、日语、汉语等。人类与计算机机对话也有许多不同语言，如BASIC，FORTRAN，COBOL和PASCAL等。在这些计算机语言中，BASIC语言在语法构造上与人类语言(英语)很接近。因此是很容易弄懂和掌握的。

1·3、2　什么是“纯净电脑”？

MZ—700是一部纯净电脑。所谓纯净，就是打开电源，电脑主存贮器是一片空白，用户必须把解释程序载入后才能使用电脑。这样虽麻烦，但也有好处，它方便你用各种不同的语言载入电脑内，这样便扩大了这部机应用软件的领域，扩大了它的使用范围。

另方面，如果一开机就是BASIC语言，这就妨碍了其它语言载入主存贮器中了。

1·3、3　把BASIC载入MZ—700

在接通电源使用MZ—700做任何工作之前，必须把BASIC解释程序载入MZ—700中。MZ—700的包装箱内附有一盒BASIC解释程序的磁带。使用BASIC解释程序载入计算机的步骤如下：

（1）按90　页的说明连接显示器。

（2）打开机后面的电源开关。

（3）此时在屏幕上将显示出如下字样和闪烁的方框图案，这图案就是光标。

```
**  MONITOR  1Z—013A  **
*▣
 └──光标
```

（4）把BASIC解释程序磁带放入盒式磁带机中。

（5）键入LOAD并按 CR 键，在屏幕上便显示出↓PLAY 。

（6）按下录音机上的PLAY按纽，盒式磁带开始旋转，说明BASIC解释程序开始载入电脑。

载入完成之后，显示READY信息且光标在下一行的第一位置上闪烁。此时屏幕上显示出：

```
** MONITOR  1Z—013A **
*   LOAD
↓   PLAY
    LOADING   BASIC
BASIC INTERPRETER 1Z—013B VX . XX
COPYRIGHT  1983 BY  SHARP  CORP
XXXXX  BYTES
READY
⊡
```

载入BASIC解释程序后，你就可以使用BASIC与电脑交谈了。

注意：

（1）LOAD指令是把磁带中录制的程序或数据载入计算机。

（2） CR 是代表回车键，主要用途是指示输入指令结束。

1·3、4　尝试执行一个程序

仅仅把BASIC解释程序载入计算机，还不能使计算机做任何工作，必须给电脑一些指令。它才会按你的意思去做。虽然我们暂时还不能解释清BASIC指令含义，但我们可以先尝试一下怎样执行一个程序。

把BASIC解释程序磁带拿出来，反转至B面朝上再放入磁带机中，有三个现成的应用程序按下面步骤把应用程序调入计算机中。

（1）把磁带反过来后，重新放入录音机内，按REWIND按扭，倒回磁带，然后按LOAD和 CR 键当屏幕显示↓PLAY信息时，在录音机上按下PLAY按扭，应用程序便开始载入电脑。

（2）当载入完成后，磁带停止转动，屏幕上出现READY信息，光标在下一行第一位置闪烁。

（3）程序载入计算机内存后，按RUN和 CR 键，程序便运行。

（4）要中断程序，同时按下 SHIFT 和 BKEAK 键，屏幕上显示出BREAK和READY字样，此时，光标在下行的开始位置内闪动。

（5）键入LIST后按 CR 键，原程序会列表显示出来，按空白键会暂停列表

（6）如果你想恢复执行程序，只要再键入RUN然后再按一下 CR 键即可。

（7）如果你想运行其它程序，可按上述步骤进行，你可把含有这个程序的盒式磁带放入录音机中，然后用LOAD命令载入程序，用RUN命令执行它。当你载入新的程序时，先前在计算机内存中的程序会自动清除。这样使计算机内存只保留BASIC的解释程序和最后一次输入的程序。

敬　告　读　者

北京市海淀区三来电子公司是北京市新技术企业开发试验区所认定的高科技公司。我公司以普及推广家用电脑为主营方向，兼营其它计算机产品、机电产品及办公自动化用品。在原中国电子学会电子服务中心的基础上，在《无线电》、《电子世界》、《家用电器》等三刊的大力协助下，三来公司将竭诚为广大电子爱好者、微机爱好者服务，努力办成我国家用电脑的窗口，与社会各界建立广泛的联系，愿成为电子爱好者的朋友。

公司地址：北京市海淀区花园路甲13号
电　话：202．5211
　　　　201．8833转344
单位全称：北京市海淀区三来电子公司
开户银行：北京市工商行北太平庄办事处
帐　号：661147－05
乘车路线：乘地铁到积水潭下车换乘331路
　　　　汽车到花园路下车。

欢迎广大用户惠顾！

三来电子公司

第二章　BASIC语言

第一节　BASIC程序设计入门

2·1、1　直接方式

在你学习程序之前，你完全可以把MZ—700当做普通的袖珍计算器，进行所谓直接方式的计算练习，如下例，依顺序键入：

$\boxed{4}+\boxed{9}=\ \boxed{CR}$

正如你所看到的，当你按上述方式键入时，计算机什么都不做。在这方面计算机与普通袖珍计算器是完全不同的，也就是说你必须按计算机的BASIC解释程序所能理解的格式键入指令，像上述问题，应用如下方式键入：

PRINT　4 + 9 $\boxed{CR}$

此时屏幕作如下显示，其中数字“13”就是正确答案。

READY
PRINT　4 + 9（PRINT是告诉计算机在屏幕显示并对其后面的表达式进行计算，
13　　　　　　　　　　这里是显示4 + 9之和。）
READY

现在让我们用直接方式作一些更为复杂的计算。MZ—700的BASIC语言中使用的基本算术运算有：

加　＋
减　－
乘　＊
除　／
乘方↑

当＊＋和↑等运算符混合在一个单一的算术表达式中时，它们的运算顺序是按运算符的优先次序进行的，正如普通的代数运算那样，运算式可以用园括号括起来，这样计算机会首先运算园括号内的式子。括号内的式子中先运算乘方，接着是乘除（它们的优先权相等，因此，运算时是从左到右，遇到乘就执行乘法，遇到除就执行除法），然后是＋和－运算。

例如要计算式子3 × 6 ×｛6 + 3 ×〔9 − 2 ×（4 − 2）+ 1〕｝

按键顺序如下：

$\boxed{P}\boxed{R}\boxed{I}\boxed{N}\boxed{T}\ \boxed{3}\boxed{*}\boxed{6}\boxed{*}\boxed{(}\boxed{6}\boxed{+}\boxed{3}\boxed{*}\boxed{(}\boxed{9}\boxed{-}\boxed{2}\boxed{*}\boxed{(}\boxed{4}\boxed{-}\boxed{2}\boxed{)}\boxed{+}\boxed{1}\boxed{)}\boxed{)}\boxed{CR}$

现在我们可以试一下，利用计算机做一些更复杂的算术运算：

练习式	在计算机键盘上键入方式及显示的答案
1. $\frac{6+4}{6-4}$	`PRINT  (6+4)/(6-4)` `5`
2. $3\times[5+9\times(9-2)-\frac{6}{4-2}]+5$	`PRINT  3*(5+9*(9-2)-6/(4-2)) +5` `200`
3. $(3+4)\times(5+6)$	`PRINT  (3+4)*(5+6)` `77`
4. $\frac{10+20}{6}\times(2+3)$	`PRINT  (10+20) /6*(2+3)` `25`
5. $\frac{10+20}{6\times(2+3)}$	`PRINT  (10+20) /(6*(2+3))` `1`

在做完以上式子以后，你可试一下键入 $\boxed{?}\boxed{5}\ \boxed{*}\boxed{8}$和按$\boxed{CR}$键，屏幕会显示答案40，这说明BASIC解释程序中指令PNINT和问号（？）的作用是相同的，所以，你应记住问号“？”是指令PRINT的缩写格式。这对于编写程序和操作都是十分方便的。

现在你可以试着键入下式（通过同时按下$\boxed{SHIFT}$ 和数字键$\boxed{2}$则键入的就是双引号）。

PRINT “4＋9＝ ”$\boxed{CR}$

正如你所看到的，在双引号之间的字符被显示在屏幕上，但它并不是答案，现请你键入下式：

PRINT “ABCDEFG”$\boxed{CR}$

这时在屏幕上将会显示ABCDEFG字样，换句话说，利用PRINT指令和双引号告诉计算机，把引号之间的内容照原样地显示在屏幕上。在这里引号内的字符是被当做字符串看待的。

再键入下式：

PRINT “4＋9＝”；　4＋9 $\boxed{CR}$

在屏幕上将出现如下显示

4＋9＝␣13……（“␣”符号表示一个空格。实际上显视屏上，这个符号所在的位置没有任何显示。）换言之，上面的指令形式是告诉计算机把引号之间的字串“4＋9＝”和算术表达式“4＋9　”结果显示在屏幕上。又键入下式：

PRINT “4＋9＝”，4＋9 $\boxed{CR}$

此时显示：

4＋9＝ --------- 13

显然这次的显示格式与上面不同，这是因为PRINT指令在显示信息项目（字串和算术式的结果）时，是根据信息项目之间用的分隔符是分号（；）或者逗号（，）而有所区别。

分号（：）是命令机算机显示项目时彼此之间以紧凑格式显示，逗号（，）是命令计算机每个显示项目占10个空格位置。如果你有一台MZ—731（或一台单独的彩色绘图打

印机）请试一试，在PRINT 指令之后挂上字符（／P）

```
PRINT／P "4＋9＝"；  4＋9 CR
```

这时，屏幕上不显示它，但从绘图打印机上打印出结果来。换句话说，符号（／P）是把屏幕显示变为打印机输出。

以上解释，就是应用袖珍计算器MZ—731的整个过程，实际上这就是MZ—700的“直接方式”。

注：PRINT“5＋8＝”；5＋8 显示 5＋8＝␣13，而PRINT“5－8”；5－8 则显示 5－8＝－3

不难看出，不论结果是正还是负，为了指示符号位总是留出一个空格位置，但只有结果是负时“－”号才占用空格，否则是空着的。

2·1、2 程序设计

试编一个简单程序，程序输入计算机之前，务必清内存，即首先键入NEW然后按下 CR 键，（关于这个指令更详细说明见18 页。）

正确地键入如下所示的程序

```
10  A＝3  CR  把 3 赋给 A
20  B＝6  CR  把 6 赋给 B
30  C＝A＋B  CR  把 A＋B的和赋给 C
40  ？  C  CR  显示 C的值
50  END  CR  表示程序结束的指令
```

每行左端的数字 10，20，30 等是程序行号亦叫行号，行号是用来说明计算机执行指令组的顺序的，先执行最小行号指令，然后执行下一行。行号必须是 1 ～ 65535 之间的整数。

在这个程序中行号可以用 1 、2 、3 等等来代替 10 、20 、30 等。但通常以 10 为间隔来编行，其目的便于插入其它程序行。

现在让我们检查一下所键入的程序是否正确，为此可键入 LIST 然后按下 CR 键，程序将一行接一行地显示在屏幕上。值得注意的是，在 40 行中开始时键入的问号（？），在计算机内已被转换成 PRINT 显示在屏幕上。为了在屏幕上显示资料，使用全屏格式命令。

```
LIST
10   A＝3
20   B＝6
30   C＝A＋B
40   PRINT  C
50   END
```

```
READY
▒
```

让我们执行这个程序

```
RUN CR
```

键入RUN和按 CR 键，在屏幕的第9行上显示出计算结果。

现在我们来解释修改程序的过程，首先，让我们把20行指令由B＝6改为B＝8，为此，我们键入LIST 20和按下 CR 键，此时，程序中的20行将显示在屏幕上，随后利用光标控制键（在键盘的右边有箭头标记的键），把光标移到数字6上，为了修改，按数字8和 CR 键。注意在修改某程序行时一定要把要修改的内容都修改了才能按 CR 键。为了确定修改是否对了，你可以再次键入LIST和按下 CR 键。

下面让我们把30行改为C＝30＊A＋B，利用光标控制键，把光标移到30行“A”的位置上，然后连续地按三次 INST 键，这就把“A＋B”向右移动三个字符位置，此时在屏幕上30行的显示如下：

```
30 C＝_ _ _A＋B
         ↑_ 光标位置
```

为了完成插入“30 ＊”，键入30 ＊和按 CR 键，可利用程序列表，检查修改是否完全正确。

我们再次修改30行，要求用“C＝30 ＊A”取代“C＝30 ＊A＋B”。为此，必须把光标移到B的右边，紧靠着B的位置上，然后按二次 DEL ，就删去“＋B”再按 CR 键，就可以了。

现用程序列表（LIST）以证实修改后的程序是否和如下所示一样

```
10   A＝3
20   B＝8
30   C＝30 ＊ A
40   PRINT  C
50   END
```

为了删掉一个完整的程序行，只要简单地键入该程序行的行号，并按下 CR 键即可。用程序列表命令作检查，便可证明该行确实被删除掉了。

另外，需要在30与40行之间插入“？A”指令时，只要键入35？A并按下 CR 键即可。然后用程序列表指令来证实该行是否已被加入程序中。再通过键入35及按 CR 键，把35行删掉。

在一个程序中用这种方法修改和插入程序行称为程序编辑，由这个过程产生的这种程序称为BASIC文本。每个程序行，包括行号，最多不得多于255个字符，倘若编写程序时用问号“？”代替PRINT指令的话，最大长度应减少4个字符。

经过上述编辑后，此时保留在计算机内存中的程序应为如下所示：

```
10 A=3
30 C=30*A
40 PRINT C
50 END
```

接着我们来解释，把计算机内存中的程序录到盒式磁带上的过程，予备一盒空白带（即未录制过资料的磁带）并把它放入磁带机中，然后从键盘上顺序地键入下式：

SAVE "CALCULATION" ↲

这里的"CALCULATION"是给录在盒式磁带中的程序赋于的程序名，程序名可以任意指定，但长度不得多于16个字符。

注：例中符号"↲"代表 CR 键。

当 CR 键被按下时，在屏幕上将显示"↓ RECORD·PLAY"等字样，意思请你按下磁带机上的 RECORD 按扭，程序将被录制到盒式磁带上。

赋给程序的名字称为文件名，规定文件名这件事并不是绝对需要的，但从文件管理观点看这样做倒是一个好主意。当然，文件名是与程序一块录制到磁带上的。

当录制完成后，在屏幕上显示READY字样表示录制工作已完成，此时按下磁带机上的STOP按扭，并重绕磁带。

当然这时计算机内存中的程序仍然存在，因此应键入NEW ↲ 删掉它（你可键入LIST ↲ 查看是否确已删掉）。现在我们可以利用LOAD指令把刚才录在磁带中的程序载入计算机内存。（详细说明在 2 页。）

当盒式磁带中录有多个程序时，鉴别载入那个程序的办法是通过把待载入的程序的文件名和LOAD指令一块键入来达到目的。例如：

LOAD "CALCULATION"

使用文件名，告诉计算机排斥不同文件名的程序进入计算机，如果不规定文件名（即仅仅键入LOAD ↲）那么计算机就把在磁带中最先遇到的程序载入内存。

注：当你使用的不是 MZ—731机上的磁带机时，其录制程序方法见 94 页的说明。

上面提到的LIST指令，可以在各种方式下使用，例如，在编辑 20行时，键入LIST 20 ↲，则行号 20的程序将显示在屏幕上。想把整个程序列表则键入LIST ↲。下面是列表指令的其他使用方式：

L I S T - 3 0 C R	列出 30行前的所有程序。
L I S T 3 0 - C R	列出从 30行后所有程序
L I S T 3 0 - 5 0 C R	列出行号 30— 50的所有程序行
L I S T 3 0 C R	列出第30行

当屏幕已占满，需要通过LIST指令，列出个别行进行编辑时，可同时按下 SHIFT 和 CLR 键（即 INST 键）即可清屏此时光标移到左上角（这样做并不影响内存中程序）。再键入LIST＜行号＞↲ 要编辑的程序便显示出来了。

第二节　BASIC语言程序设计

2·2、1　常数

所谓常数指的是写入程序中的一个数字或一个字符串，程序被执行时，可直接使用它，常数分为数值常数，字符串常数以及系统常数。解释如下：

数值常数

一个数值常数是一个最多有8位有效数字的数，指数常数必须在 10^{-38} ～ 10^{38} 范围内（最大范围是1.548437E－38～1.7014118E＋38

例如：

－123.4

0.789

3748.0

3.7E＋12…………………… 3.7×10^{12}

7.65E－9…………………… 7.65×10^{-9}　E表示指数

14.8E 9…………………… 14.8×10^{9}

十六进制数，是以十六进制格式表示的数，并仅用于在LIMIT，POKE，PEEK和USR指令，作为直接内存地址时使用。十六进制数用4位数表示，且在它的前面须以美元号（＄）

例如：

LIMIT　＄BFFF

USR（＄C○○○，×＄）…………………X＄表示字符串变量

串常数

串常数是引号之间的字符和符号组成。它在程序中总是用来写标题，或作为屏幕显示或用于打印输出。在6页中出现的那组字符"4＋9"是字符常数而不是数值常数，在BASIC中，一个串常数最多可包含255个字符（注意引号不能作字串常数使用）。

例如：

"ABCDEFG"

"12345678910"

DATA ABCDEFG……………在DATA语句中的字串常数不必加引号，当然，你喜欢的话也可以加上

2·2、2　变量

变量这个词在BASIC语言中的含意与代数表达式中的含义不同。现在我们把它放在极其简单的条款中来说明它，BASIC中的变量就像是内存中存放数值的字符的"盒子"，在

BASIC中使用的变量分三类即数值变量、字串变量及系统变量

数值变量

在数值变量中贮存的仅仅是数字资料，其定义规则如下：

i）一个变量名可以用任意个字符组成，但在BASIC解释程序中定义变量名实际上头一二个字符才有效，而且第一个字符必须是字母（A—Z）而后面的字符则可用数字或字母。

ii）不能用BASIC中的命令或语句作为变量名。

正确的变量名ABC，XY，ABCD，A 12345（ABC和ABCD被认为是同一个变量。）

不正确的变量名PRINT……（PRINT是BASIC的语句）

C@……（变量名不能包含有特殊字符。）

例如：

10　A＝5……把5赋给变量A

20PRINT　A……把变量A中的值显示出来。

串变量

串变量是用来贮存字符串的变量，其定义亦遵循定义数值变量名时所规定的规则，但串变量名的末端应加上美元（$）号，以区别其它类型变量。

一个串变量最多可以贮存255个字符。未赋值串变量是空的。

在串变量的表达式中，只允许做“十”法运算。

例如：

10A$＝“ABCD”…把字串ABCD存入串变量A$中。

20B$＝“XYZ”…把字串XYZ存入串变量B$中。

30C$＝A$＋B$…把串变量A$与B$之和（ABCDXYZ）存入串变量C$中。

40PRINTC$…显示串变量C$。

系统变量

系统变量所包含的值是由BASIC解释程序自动改变的，这些系统变量是容量（SIZE）（这个变量是指明BASIC可用的当前内存量）和时钟TI$（一个6位数变量，即系统的24小时的时钟值。）

例：

10　TI$＝“013500”…把时间1：35：00A.M的值赋给TI$，并将系统时钟置于这个时间。

20　PRINT　TI$………显示系统时钟的当前值。

显示格式：

132819……表示时间是13：28：19

PRINT　SIZE、……显示剩下的内存量（也即为了增加程序行可使用的内存量。这个变量所指明值是随程序行的增加而减少的。

2·2、3数组

不妨把数组想像为计算机内存中有一个架子，架子上放着几排盒子，每个盒子代表一个变量。盒子是按顺序编号排列的。其号码叫“脚标”，用“脚标”就可识别盒子位置了。

架子上的盒子要用一个简单命令来说明它们的存在，即所谓数组说明，数组说明规定每个架子上放的盒子数，而且架子上的盒子是整数且按规定格式排列的。

一维数组可想象成架子上只有一层容纳一排盒子，而两维数组可想象架子上有多层，每层容纳一排盒子，这些盒子或变量称为数组元素。

“脚标”号码是对应数组维数的元素，例如一维数组的每个元素，是用单一的脚标来指明盒子在排列中位置的；而两维数组是通过二个“脚标”来识别的，其中一脚标是指明排号，而另一个脚标则是用来指明该排的位置。脚标的编号是从0开始的，而最大编号则取决于每维的大小而定。

一个数组的最大容量由计算机内存空间多少来决定（即根据程序大小，在数组中要存的数据量等）。虽然BASIC的语法并不限制任何使用的数组维数，但实际上，数组维数是受可用贮存数组变量的内存量限制的。

在使用任一数组元素贮存数值之前，必须进行数组说明。

例1：

```
10 DIM A（5）………说明一维数组A中有6个元素。
20 DIM X$（8）………说明为一维串数组有9个元素。
10 DIM A（5），X$（8）………功能同10行与20行。
```

例2：

```
10 DIM B（5，5）…………2维数组B共有6×6个元素。
20 DIM Y$（5，8）…… 2维串数组Y$共有6×9个元素。
10 DIM B（5，5），Y$（5，8），A（5），X$（8）………二个数组
                                                   和二个串数组
```

例3：

```
10 DIM C（3，3，3）………说明C为三维数组有4×4×4个元素
```

注：对数组说明时，每个数组必须用不同的名字。例如DIM A（5），A（6）是非法的。

试试执行下列程序并检验其结果：

```
10 DIM A（2），B$（2）
20 A（0）＝26
30 A（1）＝9
40 A（2）＝－100
50 B$（0）＝"ABC"
60 B$（1）＝"XYZ"
```

```
70 B$（2）＝"MZ—700"
80 PRINT A（1）
90 PRINT B$（2）
100 PRINT A（2）
110 PRINT B$（0）+B$（1）
120 PRINT A（0）
```

注：数组如A（5）和X$（8）中的各变量均称为数组元素。数值常量，数值变量和数值数组均称为数值表达式；而串常量，串变量和串数组均称为串表达式

2·2、4 BASIC运算

在BASIC中，算术运算与平常算术运算是有显著不同的。在BASIC中所用的各种算术运算符如下表所示.运算符组成表达式.其优先顺序见表左数字.但有括号时.括号内优先。

算术运算

	运算符	运　算	格　式
1	↑	乘幂	X↑Y（表示X^Y即X的Y次方）
2	－	负	－X
3	＊，／	乘，除	X＊Y（X的Y倍），X／Y（$\frac{X}{Y}$即X被Y除）
4	＋，－	加，减	X＋Y（X加Y），X－Y（X减Y）

例1

10 A＝3＊8／4……当几个运算优先顺序相同的运算符同时出现在一个式子里时，计算是从左到右进行，因此左边式子的结果为6

例2　10 A＝60－6＊8＋2……结果是14

20 B＝（60－6）＊8＋2……结果是434

例3

10 A＝2↑3……把2的3次方赋给A，结果是8

字串运算：通过衔接二个或更多的字串形成一个新的字串。在字串运算中只允许使用字串相加（＋）的运算符号。

PRINT "ABC"＋"DEF"↵→显示字串"ABCDEF"

2·2、5 初始设置

下面讨论在BASIC 1Z—013B启动时的初始设置状态

- 键盘

1）按正常操作模式：为字符数字

2）定义功能键

F 1 ：………… "RUN" + CHR $（13）　SHIFT + F 1 ：……… "CHR $("
F 1 ：……… "LIST"　SHIFT + F 2 ：……… "DEF KEY("
F 3 ：……… "AUTO"　SHIFT + F 3 ：……… "CONT"
F 4 ：……… "RENUM"　SHIFT + F 4 ：……… "SAVE"
F 5 ：……… "COLOR"　SHIFT + F 5 ：……… "LOAD"

注：按定义功能键F 1操作时，已含有回车码。

- 内部时钟

系统变量TI$ 初始值是 "000000"

- 音乐功能

1）音乐完成节拍：4（适中，大约中速）

2）音符延续时间：5（1／4音符♪）

- 控制键及控制字符

当与 CTRL 键一块按下时，控制键执行特殊功能，这些键的功能以及它们对应的ASCII代码列表如下：

控制代码

CTRL＋ASCII代码（十进制数）		功能
E	5	对字母数字符，选择小写字符输出
F	6	对字母数字符，选择大写字符输出
M	13	回车（CR）
P	16	与 DEL 键同
Q	17	把光标移到下一行（↓）
R	18	把光标移到上一行（↑）
S	19	把光标向右移一个字符位置（→）
T	20	把光标向左移一个字符位置（←）
U	21	把光标移到左上角（HOME）
V	22	清屏幕为背景色（CLR）
W	23	把计算机置于绘图输入方式（GRAPH）
X	24	插入一个空格（INST）
Y	25	把计算机置于字符数字输入方式

- 其它

BASIC文本区的最低限地址是＄FEFF，这与执行LIMIT MAX是相同的

设置打印机初始状态，参考有关打印机的叙述。

第三节　常用的BASIC命令和语句

2·3、1程序文本的输入／输出指令

2·3、1.1　LOAD……（缩写型式LO.）

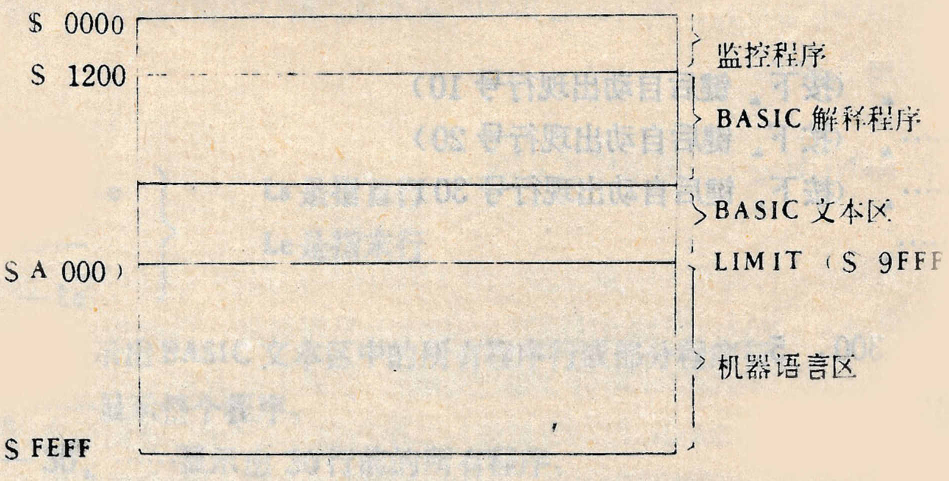
格式　LOAD或LOAD "文件名"

功能　这个命令是从盒式磁带中载入指定的BASIC文本文件或者载入一个与BASIC程序连接的一个机器语言文件（参考 2 和 9 页）

用这个命令时，只能载入BASIC文本文件或者机器语言程序。当被载入的文件是BASIC文本文件时，一旦载入新的程序，保留在BASIC文本区的程序就被清除。

注：　当载入一个机器语言子程序与BASIC程序连接时，必须执行LIMIT语句，以便在内存中留出一个机器语言程序区。此外，如果载入地址是进入机器语言区，那么一旦载完，机器语言程序就会立即执行。

载入机器语言程序文件可用LOAD指令


$ 0000
S 1200
(S A 000)
S FEFF
监控程序
BASIC 解释程序
BASIC 文本区
LIMIT (S 9FFF
机器语言区

注：BASIC文本区的最低限值可根据载入的程序文本内存容量来改变的

2·3、1.2　SAVE……（缩写格式SA.）

格式　SAVE或SAVE "文件名"

功能　这个命令将计算机内存中的BASIC程序设定一个文件名，并且把它保存在盒式磁带中

注：这个命令仅能保留BASIC程序文件（即可通过执行LIST命令显示程序文本），它不能将保留在机器语言区的任何机器语言程序文件名称与BASIC文本文件一块被录在磁带上。

规定描述文件名最多16个字符。如果没有规定文件名，录制的子程序也没有名字。然而，值得注意的是，当一条磁带中录制了许多无程序文件名的程序时，对文件管理便产生困难。

2·3、1.3　VERIFY……（缩写格式V.）校验命令

格式：VERIFY或VERIFY“文件名”

功能：这个命令是严格地校验由SAVE命令录在磁带上的程序，校验是通过放磁带中程序和内存中的相应程序进行比较，如果两个程序相同就显示“OK”字样，如果不同就显示“READ error”（读错误）如有错，应重新录制该程序。

2·3、2 文本编辑命令

2·3、2.1 AUTO……（缩写格式：A.）

格式：AUTO或AUTO Ls，n

Ls……开始行号

n……编行间隔（行号增量）

功能：这个命令是在输入BASIC程序语句时，自动地产生程序行编号。

例1：

```
AUTO     ↙  （按下↙ 键后自动出现行号10）
 10……   ↙  （按下↙ 键后自动出现行号20）
 20……   ↙  （按下↙ 键后自动出现行号30）
 30……   ↙
```

例2：

```
AUTO   300，5↙
 300……↙
 305……↙
 310……↙
```

上面命令“AUTO　300，5↙”自动设置首行为300，程序行间隔为5.

例3：

```
AUTO   100   ↙
 100…………  ↙  ┐
 110…………  ↙  ├  首行号为100，程序行间隔为10
 120…………  ↙  ┘
```

例4：

```
AUTO，20   ↙
```

10………▲
30………▲　　首行号为10 ，程序行号间隔为20
50………▲

注：按下 SHIFT 和 BREAK 键，结束AUTO命令。

2·3、2.2 DELETE……（缩写格式：D.）删改命令

格式：DELETE　Ls—Le……删去程序中Ls—Le行
　　　DELETE　　—Le……删去从首行到Le行的所有程序。
　　　DELETE　Ls —　删去从Ls行到程序结束的所有程序。

例1：
　　　DELETE　150—350▲ 删去150—350行的所有程序。

例2：
　　　DELETE—100▲ …… 100行前的所有程序删掉。

例3：
　　　DELETE　400—▲ ……从400行开始全部删掉。

2·3、2.3 LIST……（缩写格式：L.）

格式：

LIST
LIST　Ls—Le　　Ls是指首行
LIST　Ls—　　　Le是指末行
LIST　—Le

功能：在屏幕上显示出BASIC文本区中的所有程序行或部分程序行。
　　　LIST▲ ……显示整个程序。
　　　LIST—30▲ ……显示出30行前的所有程序。
　　　LIST 30—▲ ……显示出30行后的所有程序。
　　　LIST 30—50▲ ……显示行号为30至50的所有程序。
　　　LIST 30▲ ……显示30行程序。

如果想临时停止屏幕上的程序列表输出，只要按下棒状空格键即可，恢复（即放开手不按）空格键后，将重新开始继续列表输出，为了停止列表输出，只要同时按下 SHIFT 和 BREAK 键即可

2·3、2.4 LIST／P……（缩写格式L.／P）

格式：LIST／P（Ls—Le）

Ls……首行号

Le……末行号

功能：这个命令是在打印机上打印出BASIC文本区中的全部程序或部分程序行，打印出程序行的范围，与上面讨论的LIST命令相同。

2·3、2.5 MERGE……（缩写格式ME．）

格式：MERGE或MERGE "文件名"

功能：MERGE命令是用来从磁带中读出一个程序。当用这个命令读程序时，程序就被读入内存。如果忽略名字，计算机就读入在磁带中首先遇到的文件，如果读出的程序的任一行号与内存中的程序的行号相同时，在内存中的相应的程序行就被新读入的程序所取代。

2·3、2.6 NEW

功能：NEW命令是清BASIC文本区的所有变量。如果你希望输入其它程序之前，清除内存中的程序，就执行这个命令，这个命令不能清除由LIMIT命令保留的机器语言区。

由于使用LOAD命令时，能自动清BASIC文本区，所以从磁带中载入一个BASIC程序时，不必使用这个命令。

2·3、2.7 RENUM……（缩写格式REN．）

格式：RENUM

RENUM　Ln

RENUM　Ln，Lo，n

Ln新行号

Lo旧行号

n增量（编行间隔）

功能：这个命令是给一个BASIC程序重新编行，执行这个命令时，遇到分支语句如GOTO，GOSUB，ON～GOTO以及ON～GOSUB等语句其行号亦重新指定。

RENUM ……………………给当前内存中的程序重新编行，结果起始行号是10，行间间隔也是10。

RENUM　100……………给当前内存中程序重新编行，结果初始行号是100，行间间隔是10。

RENUM　100、50、20……给当前内存中的程序重新编行原来起始行号为50，重编为100行间间隔为20。

例：下面的例子是一个简单的程序执行RENUM　100、50、20的结果

重编行号之前	重编行号之后
50　A＝1	100A＝1
60　A＝A＋1	120A＝A＋1
70　PRINT A	140PRINTA
80　GOTO 60	160GOTO 120

注：当说明新的行号和旧的行号时，规定的新行号必须大于旧行号。注意，如果执行这个命令结果产生一个大于65535的行号时，将出现错误。

2·3、3控制命令

2·3、3.1 RUN……（缩写格式：R.）

格式：RUN 或 RUN Ls

Ls 起始行号

功能：执行这个命令运行BASIC文本区中的当前程序。如果从程序的第一行开始执行，那么只要键入RUN和按 CR 键即可。如果是从某一行号而不是从起始行号执行的话，那就键入RUN Ls（这里的Ls是指开始执行的行号）和按 CR 键。

执行这个命令时，在 BASIC程序还未执行前，BASIC解释程序先清除内存中所有变量和数组。）

2·3、3.2 CONT

格式：CONT

功能：按中断（BREAK）键或由（STOP）语句造成程序中断执行，使用CONT命令可以恢复程序继续执行。同样适用于END语句造成中断执行程序，但应注意END语句的后续程序行不应该是子程序。下表中列出CONT语句的适用情况：

可以继续执行程序	不能继续执行程序
·按 SHIFT 和 BREAK 键，造成程序中断， ·由STOP语句中断 ·当程序遇一个INPUT语句处于等待输入时，通过按 SHIFT 和 BREAK 键而中断程序。	·未执行RUN命令前 ·在程序执行中，由于出现错误而显示“READY” ·由于按 SHIFT 和 BREAK 键而中断盒式磁带运行， ·在执行一个MUSIC语句时程序停止执行 ·执行END语句后屏幕上显示出“READY”

2·3、3. 3 BYE……（缩写格式：B .）

格式：BYE

功能：该命令使计算机从BASIC解释程序（1Z--013B）状态进入RAM中的监控程序状态（监控命令见85页）

2·3、3. 4 KEY LIST……（缩写格式：K, L .）

格式：KEY LIST

功能：列出所有自定义功能键的内容。

```
KEY  LIST
DEF  KEY (1) = "RUN" + CHR$ (13)
DEF  KEY (2) = "LIST"
DEF  KEY (3) = "AUTO"
DEF  KEY (4) = "RENUM"
DEF  KEY (5) = "COLOR"
DEF  KEY (6) = "CHR$("
DEF  KEY (7) = "DEF KEY("
DEF  KEY (8) = "CONT"
DEF KEY (9) = "SAVE"
DEF  KEY (10) = "LOAD"
READY
▨
```

2·3、4 赋值语句　LET

格式：LET V = e或V = e

V ……数值变量或数组元素，或者字串变量或字串数组元素。

e ……数字表达式（含有一个或更多常数，变量，或数组元素）或串表达式（由一个或更多串常数，串变量或串数组元素组成）。

功能：该语句是把e的值（数值或字串）赋给等号左边的变量或数组元素V，例子如下面，LET可以省略。

例：

```
10A = 10         10LETA= 10
20B = 20         20LETA= 20
30A =A + B       30LETA=A + B
40PRINTA         40PRINTA
```

两个程序执行结果是相同的。

```
50 END                   50 END
RUN ↵
30
```

下面是不正确地使用LET语句的例子

20 A$ = A + B ……………不对，因为等号两边的变量类型不同。

30 LOG（LK）= LK + 1 ……不对　因为等号的左边既不是变量也不是数组元素

2·3、5 INPUT／OUTPUT语句（输入／输出语句）

输入／输出（INPUT／OUTPUT）语句的作用是把数据送入计算机处理以及把经处理的结果输出到TV显示屏或打印机去

2·3、5.1 PRINT（打印语句）

$$\text{格式:}\quad\left\{\begin{matrix}\text{PRINT}\\ ?\end{matrix}\right\}\left\{\begin{matrix}\text{变量}\\ \text{常量}\\ \text{表达式}\end{matrix}\right\}\left\langle\left[\begin{matrix};\\ ,\end{matrix}\right]\left\{\begin{matrix}\text{变量}\\ \text{常量}\\ \text{表达式}\end{matrix}\right\}\right\rangle\cdots\cdots$$

功能：**该语句将指定的变量、常量，字符串或表达式的值显示在屏幕上（如果PRINT）语句后无内容，则执行该语句时仅在屏幕中光标下移一行。**

在键入这个语句时，为了简化按键，可以问号（？）来代替按PRINT这个语句

数值显示格式有两种：实数型和指数型

实数型：

用实数型显示时，显示数值范围是 1×10^{-8} 到 1×10^{8}

－1.9999

63598757

0.00000001 ……… 1×10^{-8}

99999999

指数型：

用实数型不能显示的数，可使用指数型式显示

·31415 E＋9 ……… $-0.31415\times10^{+9}$

.513606 E－20 …… 0.513606×10^{-20}

用指数型或显示数值时，一个正号（＋）或负号（－）总是显示在指数（E后面的数）的前头

使用PRINT语句的一些特殊用法如下所示：

PRINT " $\boxed{C}$ "　清屏幕并把光标移到左上角。

PRINT " $\boxed{H}$ "　把光标右移到左上角但不清屏幕。

PRINT "[→]" 把光标右移一个字符位置。

PRINT "[←]" 把光标左移一个字符位置。

PRINT "[↑]" 把光标上移一行。

PRINT "[↓]" 把光标下移一行。

例：

PRINT "[C] [↓] [↓] [↓] [↓] A"……清屏幕，然后在第五行的开头位置显示A字样。

注：括孤{……}中项目为可选项。

为了键入专用字符控制光标，要首先按 [GRAPH] 键；这意味着把 BASIC 置于图象字符输出状态，此时光标图案变为"∷"，输入字符顺序如下：

输入字符	按键
[C]…………………………	按 [CLR] 键
[H]…………………………	按 [HOME] 键
[→]…………………………	按 [→] 键
[←]…………………………	按 [←] 键
[↑]…………………………	按 [↑] 键
[↓]…………………………	按 [↓] 键

键入一个专用字符之后，按 [ALPHA] 键则使BASIC从图案状态返回文本状态。

2·3、5. 2 PRINT USING……缩写格式（?USI.）

格式：PRINT USING "格式字串"，变量<{;,}变量……>

功能：这个语句是用规定的格式把数据显示在屏幕上。格式说明是由引号内的一个字符或字符串组成。而且格式说明是放在紧靠USING词的后面。例如：

（1）数值格式说明

（a）#

数字符号"#"是用来说明显示数据的最大位数。如果被显示数的位数小于格式说明中的"#"符的个数，则显示在规定的格式域内的数字一律向右对齐。

例：

```
10A=123
20PRINTUSING "####";A
RUN ↲
 123
```

（b）：由"#"符号组成的格式内可以包含一个小数点，用以指明在格式中小数点显示的位置。小数点右边的"#"的个数是用来说明显示小数的位数的

```
**1234
```

（f）££

一对英磅号（££）用于格式串前面，数值左边便有“£”。

例：
```
10A=123
20PRINT USING "££####"; A
RUN↲
——£ 123
```

（g）SS

在格式串的开始位置写入一对美元（S）号，可使显示数字之前显示一个美元（S）号。

例：
```
10A=123
20PRINT USING "SS####"; A
RUN↲
——S 123
```

（h）↑↑↑↑

在格式串的末端写上4个指数运算符号（↑）则数值以指数方式显示。

例：
```
10A=51123
20PRINT USING "##.###↑↑↑↑"; A
RUN↲
—5.112E+04
```

注意：在这种情况下，应在前头多留一个数值符号位。

（i）操作数列表

变量列表可以用一个PRINT USING语句来说明，变量之间可用逗号（，）或分号（；）作为它们之间的分隔符，结果显示按格式串规定的格式表示。

例：
```
10A=5.3:B=6.9:C=7123
20PRINT USING "##.###"; A ;B ;C
RUN↲
— 5.300—6. 900— 7.123
```

（2）字符串格式说明

（a）感叹号（！）作为格式串，仅显示其第一个字符

例：
```
10A$="CDE"
```

```
20PRINT USING "!"; A$
RUN ↵
C
```

（b）& ————— &

用"&—————&"作为格式串，则显示从头算起（2＋n）个空格的字符，（n为格式串中空格—的个数）少于（2＋n）个字符，则显示字符串本身。

例：

```
10A$ = "ABCDEFGH"
20PRINT USING "&— — — — &"; A$
RUN ↵
ABCDEF
10A$ = "XY"
20PRINT USING "&— — — — &"; A$
RUN ↵
XY
```

（3）字串常量输出功能

在PRINT USING语句中，任何字符均可放在格式串前面，用分号分隔开格式串与数值，则这些字符与数值一齐显示。

例：

```
10A= 123
20PRINT USING "DATA####"; A
RUN ↵
DATA— 123
```

（4）USING的分隔

通常PRINT与USING是连在一起使用的，但在同一语句行中也可分开使用。

例：

```
10A=-12: B= 14: C= 12
20PRINTA; B; USING "####"; C
  \______ ______/ \______ ______/
         V               V
   正常打印功能       USING功能
RUN ↵
-12— 14—— 12
```

2·3、5.3 INPUT（输入语句）（缩写：I.）

格式：

$$\text{INPUT}\left\{\begin{array}{l}\text{数值变量}\\ \text{字串变量}\\ \text{数组元素}\end{array}\right\}\cdots\cdots\text{或 INPUT “字符串”；}\left\{\begin{array}{l}\text{数值变量}\\ \text{字串变量}\\ \text{数组元素}\end{array}\right\}\cdots\cdots$$

```
INPUT A          INPUT "DATA A="; A
INPUT B$         INPUT "YESOR NO"; B$
INPUT X(5)       INPUT "KEYIN"; X(5)
```

功能：

赋值

INPUT是一个语句。它的作用是在执行程序时，要求用户从键盘中给指定的变量。赋值执行中的程序遇到INPUT语句时就暂停执行，以使用者从键盘中打入数值。输入完成后，随即按 CR 键，此时输入的值就被赋给指定的变量，然后程序恢复执行。

例：

```
10INPUT A, B
20C=A+B
30PRINTC
40END
```

当执行上述程序时，屏幕上就会显示出一个问号，且光标不断闪动，表示计算机在等待输入数据，键入任一个数，然后按 CR 键，此时键入的值就被赋给变量A。

接着会再次显示问号（？），这是因为在第10行的INPUT语句中，说明了两个变量（A和B）而直至现在只键入一个值（此值被赋给变量A）。现再键入任意一个值，然后后按 CR 键，这个值将被赋给变量B，并且从程序的下行开始执行，在上面的例子中后面的程序行是把A和B相加并把结果赋给C，然后打印C的值。

由于在这个例子中，使用的变量是数值变量，所以在计算机等待输入数值时，如果键入的是字符而不是数字的话，屏幕上将会显示“ILLEGAL DATA ERROR”（数据非法的错误）信息。并且重新显示问号（？），提醒使用者重新输入合法的数据（无论输入的值是赋给变量，还是数组元素，数值的键入方法都是一样的）务必注意的是输入数据的类型必须与在INPUT语句中说明的变量类型相匹配。

在程序运行时，要记住屏幕显示问号（？）后输入什么数显然是困难的。因此在INPUT语句中常常含有一个“提醒字符串”，以便显示在屏幕上作为对使用者的提示。下面所示的程序在INPUT语句中，就使用了“提醒字串”：

```
10INPUT "A="; A
20INPUT "B="; B
```

```
30PRINT "A+ B =" ; A+B
40PRINT "A-B=" ; A-B
50PRINT "A*B=" ; A*B
60PRINT "A/ B=" ; A/ B
```

试运行上面的程序就明白，在PRINT和INPUT语句中包含的字符串为程序的操作提供了清晰的提示，事实上计算机的应用程序，类似于这里的指令序列组，只不过复杂得多而已。按不同的方法不同次序组合命令和语句，你很快会发现为了达到所希望的结果有许多不同的手段。

2·3、5.4 GET语句

格式：GET V

功能：V……是数值变量或数组元素，或者串变量或字串数组元素在执行程序期间，如果遇到这个语句，BASIC解释程序将检查键盘上是否有任何键被按下？如果已按下，就把相应的值赋给变量。尽管INPUT语句在输入数据前用提示符提示，只有输入数据后，程序才恢复执行，而GET语句则不管是否按下键，程序都继续执行。

在执行GET语句时，有键按下，则该语句就把数据赋给变量V，无键按下，则变量是全空（对一个数值变量将是0而对字串变量则是空格）。

使用数值变量时，这个语句只允许键入单位数（0～9）如果是字串变量，则只允许键入单一的字符。

当你想键入数据又不按 CR 键时，这个语句格外好用，像游戏程序就常常是这样的。

例：

```
10PRINT  "NEXT GO？（Y OR N）"
20GET A$
30IF A$ = "Y" THEN  50 ………
40GOTO  20……………·
50PRINT "PROGRAM END"
60END
```

在例子中，如果字串变量A$的值是"Y"的话，就从30行跳到50行去执行，条件不成立，就转到20行去执行。

这个程行显示提示符"NEXT GO？（Y OR N）"并且为了等待输入程序在20—40循环执行，直到Y键被按下时，程序才能转到50行执行而结束。现在删去30和40行再执行程序一次，你将看到无论你是否按任何键，程序都马上执行完。

注：

当接连地执行GET语句时，在它们之间应包含一个子程序，以保证继续到下一个之前每个都完成，原因是键的振动（键的触点开关是摆动的）可能导致二个GET语句同时被执行。

2·3、5.5 READ—DATA……（缩写格式REA.—DA.）

格式：

$$\text{READ}\left\{\begin{matrix}\text{数值变量}\\ \text{字串变量}\\ \text{数组元素}\end{matrix}\right\} > \left\{\begin{matrix}\text{数值变量}\\ \text{字串变量}\\ \text{数组元素}\end{matrix}\right\},\cdots\cdots\cdots >$$

$$\text{DATA}\left\{\begin{matrix}\text{数值常数}\\ \text{串常数}\end{matrix}\right\} < \left\{\begin{matrix}\text{数值常数}\\ \text{串常数}\end{matrix}\right\}, >$$

功能：

象INPUT和GET语句一样，READ语句是把数据送入计算机处理，然而它又不象INPUT和GET语句，数据不是键盘中输入，是从程序本身的DATA语句中读数，更特别的是，READ语句的作用是从DATA语句的数据列中逐项读出送入相应的变量中去，这样做时，READ语句的变量与DATA语句的数据项必须一一对应。

例1：

```
10READ  A，B，C，D
20PRINT  A；B；C；D
30END
40DATA  10，100，50，60
RUN↵
10   100   50   60
```

在这个例子中由READ语句把DATA语句中列出的值顺序地读出并分别赋给变量A，B，C，和D，然后显示这些数值。

例2：

```
10READ  X$，A1，Z$
20PRINT  X$；A1；Z$
30END
40DATA  A，1，C
RUN↵

A␣ 1C
```

如左面例子所示，在DATA语句中的字串数据不必用引号括起来。

在这个例子中，READ语句顺序地从DATA语句的数据项中逐项读出，然后把每项代入READ语句中列出对应变量中去。

例3：

```
10DIM  A（2）
20READ A（0），A（1），A（2）
30PRINT  A（0）；A（1）；A（2）
40END
50DATA  3，4，5
RUN↵
 3   4   5
```

在这个程序中READ语句后面的数据，依次读出并赋给数组元素A（0）、A（1）和A（2）然后在30行的PRINT语句把数组元素的值显示出来。

例4：

```
10READ  A
20READ  B
30DATA  C
```

这个例子是不对的，因为①在第10行的READ语句中说明的是数值变量，而DATA语句后面的数值是字符串，②在20行的READ语句无数据可读。

2·3、5．6　RESTORE（恢复语句）……（缩写格式RES．）

格式：

RESTORE或RESTORE　Ln

功能：每执行READ语句，BASIC解释程序就将指针指向从DATA语句中下一项数据。RESTORE语句恢复指针分二种(1格式为RESTORE时，执行RESTORE语句后，指针指向第一个 DATA 语句的首项数据，也就是说其后所遇的第一个READ语句重新从数据区的第一个数据读数。(2) 格式为RESTORE　Ln时，执行这个语句后，指针指向说明行Ln的DATA语句的第一项也就是说以下所遇的第一个READ语句，将从指定行Ln的DATA语句的第一项读数。

例：

```
10 DATA 1. 2. 3
20 DATA "AA" "BB"
30 READ X. Y
40 READ Z. VS
........................................
100 RESTORE
110 READ A. B. C. D$. ES
........................................
200 READ I. J
210 RESTORE
220 READ M. N
230 RESTORE 250
240 READ O. P
250 DATA 1. 2. 3. 4
260 DATA -1. -2. -3. -4
```

如果在Ln中说明行号不存在，将会产生错误。

函数RND（X）是产生随机数（参看57页）

```
10  X=33*RND(1)
20  FOR  A=1  TO  5
30  READ  M$
40  PRINT  TAB(0);"◆":TAB(X);M$;
50  PRINT  TAB(37);"◆"
60  NEXT  A
70  Y=10*RND(1)
80  FOR  A=1  TO  Y
90  PRINT  TAB(0);"◆";
100  PRINT  TAB(37);"◆":NEXT
110  RESTORE:GOTO  10
120  DATA"  ◢○◣","▣✲✲✲▣"
130  DATA"  ✲✲✲","▣✲✲✲▣"
140  DATA"  ◥■◤"
```

注：　TAB函数见47页　FOR—NEXT语句见本页。

2·3、6．循环与分支指令

2·3、6．1　FOR—NEXT（循环语句）（缩写格式：F．—N．）

格式：

FOR　　CV＝iv　TO　fv　＜STEP SV＞

……………

NEXT＜CV＞

cv…………循环控制变量；数值变量或数组元素。

iv…………初值；一个数值表达式。

fv…………终值；一个数值表达式。

sv…………增量或步长值；一个数值表达式（如果被忽略，则步长为1）

功能：

这个语句是把FOR与NEXT之间的指令重复执行一定次数。

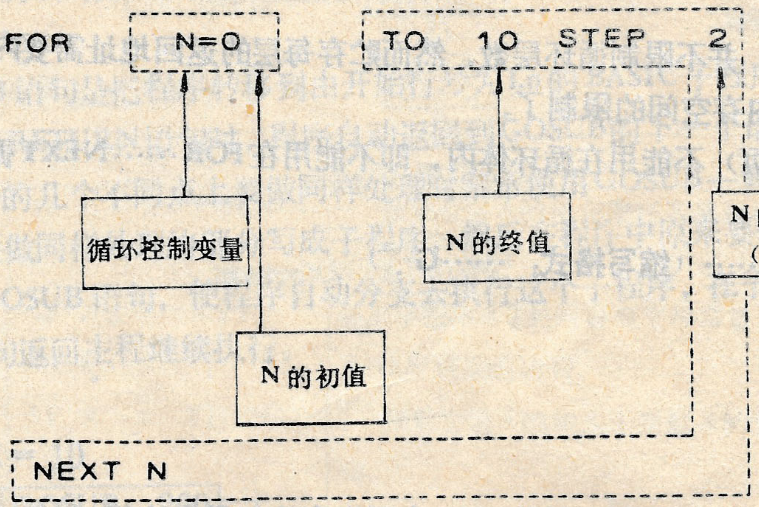
```
10A＝0
20FOR  N＝0  TO  10  STEP  2
30A＝A＋1
40PRINT"N＝"；N，
50PRINT"A＝"；A
60NEXT  N
```

（1）在上面程序中，0作为初值赋给N

（2）随后执行 30到50行，显示A· N的值。

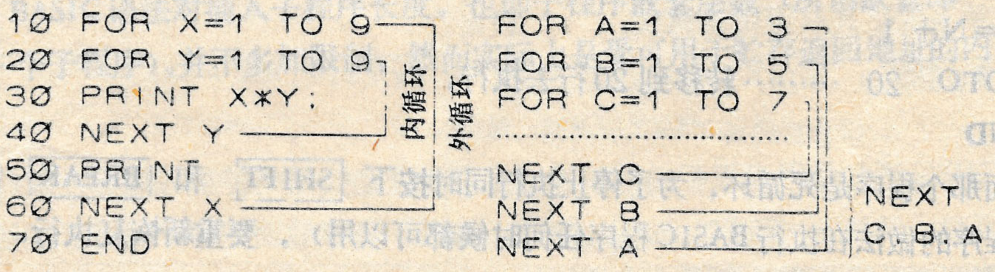
（3）在60行N的值增加2，接着BASIC解释程序检查N的值是否大于10，倘若不是便重复，执行20行以下的程序段，周而复始，当N值超过10时，执行跳出循环体并开始执行60行之后的程序行。

如< STEP SV >从语句说明中省掉，则每循环一次N的值自动增加1． 如在上面程序的情况下，如果省掉< STEP SV >则循环体内指令被重复执行11次。


FOR
N=0
TO 10 STEP 2
循环控制变量
N的初值
N的终值
N的增量
（步长）
NEXT N

FOR……NEXT循环可以插在另一个FOR……NEXT循环之中，也即允许多重循环。当需要这样做时，内层循环必须完全地被包含在外层循环之中。此外，对于每层循环必须有对应的独立的控制变量。

例：



```
10 FOR X=1 TO 9
20 FOR Y=1 TO 9
30 PRINT X*Y;
40 NEXT Y
50 PRINT
60 NEXT X
70 END
```

```
FOR A=1 TO 3
FOR B=1 TO 5
FOR C=1 TO 7
.........................
NEXT C
NEXT B
NEXT A
```

```
NEXT
C,B,A
```

当C，B，与A 循环的结束都在同一点上时，如上面程序所示，可以只用一个NEXT语句来标明所有循环结束。

病例：

```
┌─FOR J=1 TO 10
│┌─FOR J=K TO K+S
└┴─NEXT J
```

不对：在每个循环中必须使用不同的控制变量

```
┌─FOR I=1 TO 10
│┌─FOR J=K TO K+5
│└─NEXT I
└──NEXT J
```

不对，循环不能相互交叉

注：

BASIC的语法中，并不限制循环层数，然而贮存每层的返回地址需要内存空间，这样循环层数就受到可用内存空间的限制了。

CLR语句（见43页）不能用在循环体内，即不能用在FOR……NEXT循环之中。

2·3、6．2 GOTO……（缩写格式：……G．）

格式：

GOTO Ln

Ln指定行号

功能：这个语句是无条件地把程序转移到Ln中所指定的行号去执行，如果 Ln 头一行可执行（不是REM或DATA语句）则就从这一行重新开始执行，否则，执行 Ln 行的后一行。

例：

```
10N=1
20PRINT N
30N=N+1
40GOTO 20 ………转移到20行去执行。
50END
```

因为上面那个程序是死循环，为了停止执行同时按下 SHIFT 和 BREAK 键即可。（这种中断程序的做法在执行BASIC程序任何时候都可以用），要重新恢复执行．只要打入CONT）命令即可。

注意：

在GOTO语句中说明的行号，不应是FOR……NEXT中的行号，即不能从循环体外部用一个GOTO语句转入循环体内。

2·3、6．3 GOSUB—RETURN……（缩写格式：GOS·—RET·）

格式：GOSUB Ln

```
        ⋮
        ⋮
    RETURN
```

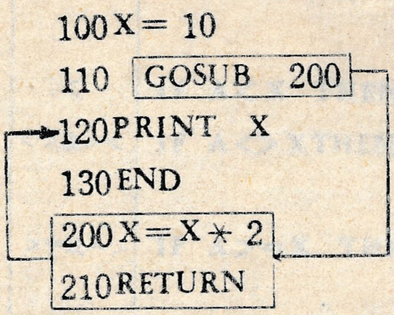
Ln 子程序开始语句的行号

功能：

GOSUB 语句是把程序转移到由开始行号为 Ln 的 BASIC 子程序去执行，子程序执行完后，在执行 RETURN 语句时，程序自动返回到 GOSUB 的下一个语句继续执行。

当在程序的几个不同点上要做同样处理时常常使用 GOSUB—RETURN 语句，在这种情况下，把要做同样处理的部分写成子程序，然后在程序中原来要做同样处理的适当位置上使用一个 GOSUB 语句，使程序自动分支去执行这个子程序，在子程序执行完成后，通过 RETURN 语句返回主程继续执行。

例：

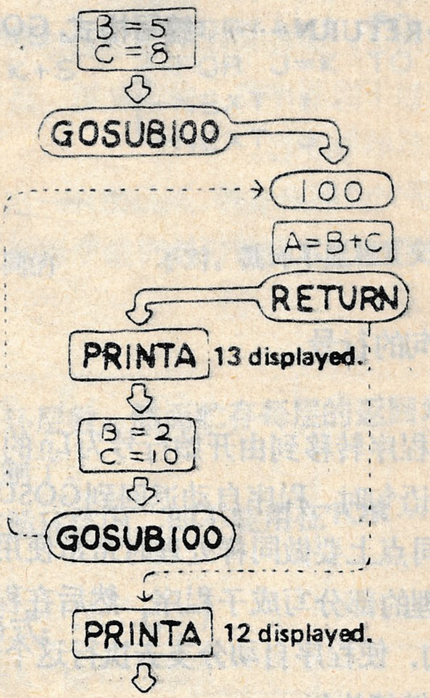


```
100X＝10
110 GOSUB  200
120PRINT  X
130END
200X＝X＊2
210RETURN
```

注：

BASIC 语法对插入子程序长度，也即子程序嵌套层数（所谓嵌套即一个子程序又包含另一个子程序），并不多加限制，然而实际上是受可用于贮存返回地址的内存空间限制的。

```
10 B=5
20 C=8
30 GOSUB 100
40 PRINT A
50 B=2
60 C=10
70 GOSUB 100
80 PRINT A
90 END
100 A=B+C
110 RETURN
```


B=5
C=8
GOSUB100
100
A=B+C
RETURN
PRINTA
13 displayed.
B=2
C=10
GOSUB100
PRINTA
12 displayed.

2·3、6. 4 IF—THEN……（缩写格式：IF—TH）

格式：

IFe THEN Ln

IFe TNEN语句

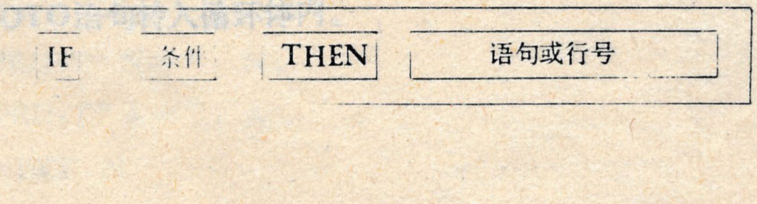
e 关系表达式或逻辑表达式

Ln 指定行号

功能：

IF……THEN语句是用来根据关系表达式或逻辑表达式结果控制程序分支执行，当表达式结果为真时，就执行 THEN 后面的语句。如果THEN后面是一个行号，表达式结果为真的话程序就自动跳到那个行号去继续执行。

如果逻辑表达式或关系表达式结果为假，程序就从IF—THEN语句所在行的下一行开始继续执行。


IF
条件
THEN
语句或行号

例：

```
IF……THEN 100
IF……THEN GOTO or IF……GOTO
IF……THEN PRINT or IF……THEN ?
IF……THEN A=5*7  赋给
IF……THEN I=10 J=50
IF……THEN INPUT
IF……THEN READ
IF……THEN GOSUB
IF……THEN RETURN
IF……THEN STOP
IF……THEN END
```

逻辑和关系表达式例子

	操作	应用实例	解释
关系表达式	=	IF A=X THEN…	如果变量A的值与变量X的值相等则执行THEN后面的语句
		IF A$="XYI" THEN…	如果字符串A$的内容与字串常量"XYZ"相同，则执行THEN后面的语句，
	>	IF A>X THEN…	如果变量A的值大于变量X的值则执行THEN后面的语句，
	<	IF A<X THEN…	如果变量A的值小于变量X的值则执行THEN后面的语句，
	<>或><	IF A<>XTHEN…	如果变量A的值不等于变量X的值，则执行THEN后面的语句，
	>=或=>	IF A>=X THEN…	如果变量A的值大于或等于X的值，则执行THEN后面的语句，
	<=或=<	IF A<=X THEN…	如果变量A的值小于或等于X的值，则执行THEN后面的语句，
逻辑表达式	*	IF(A>X)*(B>Y) THEN…	IF变量A的值大于变量X的值，以及变量B的值大于变量Y的值，则执行THEN后面的语句，
	+	IF(A>X)+(B>Y) THEN…	如果变量A的值大于变量X的值或变量B的值大于变量Y的值，则执行THEN后面的语句，

注：

用BASIC 1Z—013B进行数值比较时应小心谨慎，在计算机内部数值用二进制浮点形式表示的，为了处理和送外部显示必须把数值转换成另外形式（如用PRINT语句时换成10进制数形式）这种转换可能会产生误差。

例如：

当一个算术表达式的计算结果是一个整数时，就不必再做转换运算，如果处理后的值不是整数，那就必须进行转换运算，在使用等号"="的关系中，务必要特别注意考虑这

一点。

下面的程序例子说明，在检验1和1／100＊100的相同与否时，得到FALSE（否定的结果）

```
10A＝1／ 100＊100
20IF A＝1  THEN  PRINT  "TRUE"：GOTO  40
30PRINT  "FALSE"
40PRINT  "A＝"；A
50END


RUN
FALSE
A＝1
```

事实上这个程序结果是显示"FALSE"和"A＝1"这就说明数的内部表示和外部表示是不同的。

因此，像上面程序中为了检验相等性，最好是用下面所示方法：

```
20IF ABS（A－1）<．1E－8  THEN  PRINT  "TRUE"：
  GOTO  40
```

2·3、6．5 IF—GOTO ……（缩写格式IF—G．）

格式：

IFe GOTO Lr

e：关系表达式或者逻辑表达式

Lr：指定行号

功能：

这个语句序列是检查关系或逻辑表达式定义的条件。如果条件满足，则分支到Lr行。正如IF—THEN序列一样，IF—GOTO也是用于条件分支：如果规定的条件得到满足，程序就跳到Lr行去执行，若条件不满足，就执行下一行语句。

（在同一程序行的IF—GOTO后的任何语句均被忽略）。

例：

```
10G＝0：N＝0
20INPUT "GRADE＝"；X
30IF X＝999  GOTO  100
40T＝T＋X：N＝N＋1
50GOTO  20
100PRINT "－－－－－－－－－－－"
```

```
110PRINT  "TOTAL: " ; T
120PRINT  "NO. PEOPLE: " ; N.
130PRINT  "AVERAGE: " ; T/ N
140END
```

2·3、6．6 IF—GOSUB……（缩写格式：IF—GOS.）

格式：

IFe GOSUB Lr

e：关系表达式或逻辑表达式

Lr：指定行号

功能：

这个语句检查由关系式或逻辑表达式e定义的条件，然后，如果条件满足，就分支到Lr行开始的子程序去，子程序执行完后，返回到IF—GOSUB语句的下面第一语句继续执行，因此，如果在包含IF—GOSUB的程序行中有多个语句，则返回到IF～GOSUB语句后的第一个语句继续执行。

```
10 INPUT "  X=  ";X
20 IF X<0 GOSUB 100:PRINT X<0"
30 IF X=0 GOSUB 200:PRINT X=0"
40 IF X>0 GOSUB 300:PRINT"X>0"
50 PRINT "-------------------------"
60 GOTO 10
100 PRINT " * PROGRAM LINE 100 ":RETURN
200 PRINT " * PROGRAM LINE 200 ":RETURN
300 PRINT " * PROGRAM LINE 300 ":RETURN
```

2·3、6．7 ON—GOTO……（缩写格式：ON—G.）

格式：

ONe GOTO Lr_1 <，Lr_2，Lr_3，……，Lri >

e ……数值变量，数组元素或表达式

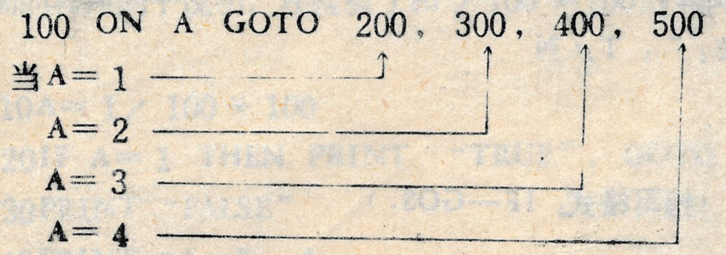
Lri 指定行号列

功能：

这个语句是多分支语句根据e的值，转移到GOTO后面行号之一中去执行。

计算机在执行这个语句时，根据e的值，到GOTO后面的行号表中去找出第e个位置上（从左算起）的行号，然后转到这个行号所指的程序行去执行，也就是说如果e是1，就转到行号表中第一个行号所指的程序中去执行，如果e是2，就转到行号表中，第二个

位置上的行号所指的程序去执行，依此类推，例如：



```
10 INPUT"NUMBER";A
20 ON A GOTO 50,60,70
50 PRINT"XXX":GOTO 10
60 PRINT"YYY":GOTO 10
70 PRINT"ZZZ":GOTO 10
RUN
NUMBER ? 1
XXX
NUMBER ? 2
YYY
NUMBER ? ■
```

如果是一个带小数的数，则在执行多分支语句之前，先把小数截去。

注：

当e的值大于GOTO后面行号表的一个数时则计算机会跳到程序的下一行开始继续执行，如果e的值小于1或负数时，计算机也是这样处理。

此外，如果e是一个非整数，则在执行该语句前，先把小数部分截去取整。

2·3、6．8　ON—GOSUB……（缩写格式：ON—GOS．）

格式：

ONe GOSUB Lr$_1$ ＜，Lr$_2$，Lr$_3$，……，Lri ＞
e……数值变量，数组元素或表达式
Lri　一组子程序开始语句的行号

功能：

这个语句也是多分支语句，这个语句被执行时，计算机根据e的值，调用 GOSUB 后面指定的某一行号的子程序，这个语句的操作基本上与ON—GOTO语句相同，但所有的分支都是子程序，子程序执行完后，由 RETURN语句从子程序返回到调用该子程序的ON—GOSUB语句后面的第一个可执行语句开始继续执行。

例：

下面程序使用了ON—GOSUB语句，其中从180行开始是个子程序，执行了190行又调用别的子程序，可见子程序可以嵌套。

```
10 A$=" ENGL ":B$=" MATH ":C$=" FREN
20 D$=" SOI ":E$=" MUS ":F$=" GYM
30 G$=" HIST ":H$=" ART ":I$=" GEOG
40 J$=" BUS ":K$=" H RM ":PRINT 🄲"
50 INPUT"WHAT DAY?";X$
60 FOR Z=1 TO 7:Y$=MID$("SUNMONTUEWEDTHU
FRISAT".1+3*(Z-1).3):IF Y$=X$ THEN X=Z
70 NEXT Z
80 FOR Y=0 TO 4:PRINT TAB(5+6*Y);Y+1;
90 NEXT Y:PRINT
100 ON X GOSUB 180.120.130.140.150.160.170
110 PRINT:GOTO 50
120 PRINT"MON ";A$;B$;D$;G$;K$:RETURN
130 PRINT"TUE ";B$;E$;H$;H$;D$:RETURN
140 PRINT"WED ";C$;C$;I$;A$;F$:RETURN
150 PRINT"THU ";B$;D$;F$;G$;E$:RETURN
160 PRINT"FRI ";A$;D$;I$;C$;C$:RETURN
170 PRINT"SAT ";B$;G$;D$;K$:RETURN
180 FOR Y=1 TO 6
190 ON Y GOSUB 120.130.140.150.160.170
200 PRINT:NEXT Y
210 RETURN
```

2·3、7 定维语句

2·3、7.1 DIM

格式：

DIM a_1 (i_1) <, a_2 (i_2) , ………………………… ai (im) >

DIM b_1 (i_1, j_1) <, b_2 (i_2, j_2) , ………………………… bi (in, jn) >

ai ……………………一维数组名

bi …………………… 二维数组名

im, in, jn, ………维数下标

功能：

这个语句用来定义一至四维的数组，通知内存留出空间，定义数组名最多只能用 2 个字符，而理论上多于三维数组同样可以使用，然而实际上常常受到可用内存空间的限制。

例:

```
10 DIM A (100)
20 FOR J=0 TO 100
30 READ A (J)
40 NEXT J
50 DATA 5. 30. 12. ……
```

例

```
10 DIM A$ (1) . B$ (1) . C$ (1)
20 FOR J=0 TO 1 : READ A$ (J) . B$ (J)
30 C$ (J) =A$ (J) + " " +B$ (J)
40 PRINT A$ (J) . B$ (J) . C$ (J)
50 NEXT J
60 END
70 DATA YOUNG. GIRL. WHITE. ROSE
```

注:

执行DIM语句，数组的所有元素便充0＜数值数组＞或全空＜字串数组＞，因此，该语句在数组赋值前执行，即先定义后使用。

定义数组时，每个数组名不应相同，例如，指令DIMA（5）A（6）是非法的。

使用CLR语句和NEW语句，能清除DIM的说明。

2·3、7．2 DEF FN

格式:

DEF FN f (X) = e

f ……由一个大写字母表示的自定义名。

X……变量名。

e ……表达式或用户自定义函数。

功能:

DEF FN语句是用户自行定义的函数语句，这个函数由BASIC固有函数组合而成。

例如:

DEF FNA（X）= 2＊×↑2＋3＊×＋1…… 定义 $2X^2+3X+1$为 FNA（X）

DEF FNE（V）= 1／2＊M＊V↑2…………… 定义 $1/2MV^2$为 FNE（V）

10DEF FNB（X）=TAN（X－PA1（1）／6）

20DEF FND（X）=FNB（X）／C+X 不对的定义，如:

10DEF FNK（X）=SIN（X／3+PAI（1）／4）. FNL（X）= EXP（－X↑2／K）

DEF仅能定义一个定义函数。

一个重量为5.5物体，其初始速度分别为 3.5，3.5×2，3.5×3，求其动能。

解题程序如下：

```
10DEF  FNE（V）＝1／2＊M＊V↑2
20M＝5．5：V＝3.5
30PRINT  FNE（V），FNE（V＊2），FNE（V＊3）
40END
```

注：

执行CLR语句和NEW语句，会清掉一切自定义函数。

2·3、7．3 DEF KEY

格式：

```
DEF KEY（K）＝S$
```

K……自定义功能键对应值（1～10）

S$……字符串＜最多15个字符）

功能：

可以把字符串赋给十个功能键中的任何键这样在任何时候只要按一个键，就能把字串送入计算机。这个语句就是定义这样的字串，并且把它们赋给可定义的功能键。要键入1～5的功能键，只要按键盘左上面的对应键即可，但要键入功能键6～10时，则要同时按下 $\boxed{\text{SHIFT}}$ 和对应的功能键才成。在K中说明功能键编号而S$是代表赋给功能键的字串或命令。执行DEF KEY语句，就取消先前对可定义的功能键的定义。在同一行中，一个DEF KEY语句之后，不能有别的语句。

例：

10DEF KEY（1）＝“INPUT”………定义功能键 $\boxed{F_1}$ 为INPUT

20DEF KEY（2）＝“RUN”＋CHR$（13）………定义功能键 $\boxed{F_2}$ 为RUN。

注：

CHR$（13）是 $\boxed{\text{CR}}$ 的ASCII代码。把它和字串一块赋予一个可定义的功能键有如同按 $\boxed{\text{CR}}$ 键的效果（CHR$的功能讨论在61页，而ASCII代码表在132页。）

2·3、8 注释语句和控制命令

2·3、8．1 注释语句（REM）

格式：

REMr

r……程序的注释

功能：

REM注释语句是一个非执行语句，占一个程序行，但BASIC解释程序不予记忆，因为REM是一个非执行语句，因此它可以放在程序的任何位置而不影响执行结果。为了阅读程序方便，使用REM语句，给程序注释。

多语句程序行

在一个程序行中包含多个语句时，语句之间必须用冒号（：）分隔开，对多语句行，BASIC解释程序的操作如同把这些语句写在不同行上的情况是一样的，例如下面二个程序得到结果是完全一样的。

```
10A=5      ┐
20B=8      │
           ├──→ 10A=5: B=8: C=A*B: PRINTC
30C=A*B    │
40PRINTC   ┘
```

注：

但也要注意，当多个语句行被用在如下所示情况时程序作运算可能不同。

```
10INPUT A
20B=0
30IF  99<A  THEN  B=1
40PRINT  B
50END
```

如果在10行输入的值大于或等于100时，这个程序运行结果显示1。IF输入值小于100则显示0。

```
10INPUT  A: B=0: IF  99<A  THEN  B=1: PRINT  P
20END
```

如果输入的值大于或等于100时，这个程序也显示1，但如果输入的值小于100时，什么结果也不显示，这是因为IF的条件不满足时，THEN后面的语句不能执行。

2·3、8．2 STOP（暂停语句）……（缩写格式：S．）

格式：

STOP

功能：

临时停止程序执行并显示BREAK和READY，然后处于等待用直接方式输入执行命令。

STOP语句用于临时中断程序执行，而且可以在程序任何需要地方都可以插入这个语句

由于程序的执行仅仅是临时中断，此时你可以以直接方式使用 PRINT 语句来检查变量贮存的值。这在调试程序时是很有用的。之后，你可以通过打入CONT命令恢复程序执行。

例：

```
10READ  A, B
20X=A*B
30STOP
40Y=A/B
50PRINT  X, Y
60DATA  15, 5
70END
RUN
BREAK  IN  30
```

2·3、8. 3　END……（缩写格式：E.）

格式：

END

功能：

END语句结束程序执行，BASIC解释程序返回到直接输入命令状态。当这个语句被执行后，将在屏幕上显示“READY”字样，以表示BASIC解释程序处于等待命令状态。整个程序执行完后，即使END设置在程序中间，也不能用CONT语句恢复。

注：

当END语句执行完，全部打开的文件便关闭。

STOP与END语句不同点：

	屏幕显示	文件	执行恢复
STOP	BREAK IN XXX READY	打开文件不关闭	用CONT能恢复
END	READY	打开文件关闭	不能恢复

2·3、8. 4　CLR

格式：

CLR

功能：

CLR命令清除所有变量，并抹去所有数组说明：所有数值变量清0，而所有字串变量全空；先前由DIM语句定义的数组全部抹掉（删掉），再次使用数组之前，

必须用DIM语句重新定维。

CLR命令也会抹去所有用DEF　FN语句定义的函数，因此，在再要使用它们之前也必须用DEF　FN语句重新定义。

注：

CLR语句不能包含在FOR—NEXT循环之内或者BASIC子程序之中。

2·3、8．5　TI$

格式：

TI$“hh mm ss”

功能：

TI$是系统字串变量名，用于设置计算机固有的时钟的时间。

这个内部变量从建立的时候起，自动地一秒一秒地增加的方式来计时，在这个变量中的6个字符，是表示时，分，秒的意思。（各用二个字符表示）。例如，如果在TI$串变量内包含的字符是“092035”时，表示时间是9：20：35A．M（即上午9时20分35秒）。

当BASIC解释程序载入计算机时，变量TI$自动置为00：00：00。为了设置今天时间可使用字串（赋值）语句。例如，通过执行如下语句，可以把时钟设置为7：00：00P·M·（即下午七时整），TI$＝“190000”

把时钟设置为7：00：00，然后按 CR 键，内部时钟即开始自动计时，设置时间时，小时数必须在0～23范围内，而分、秒数则必须在0～59之间。

下面程序是显示世界上一些城市的当前时间：

```
10 PRINT "C"
20 DIM C$(10),D(10) E(10),T$(10)
30 FOR I=1 TO 10:READ C$(I),D(I):NEXT I
40 PRINT"ENTER NEW YORK TIME (HOUR,MINUT
E,SECOND)"
50 INPUT. B$:TI$=B$:PRINT"C"
60 PRINT "H":T$(1)=TI$
70 FOR I=1 TO 10
80 E(I)=VAL(LEFT$(T$(1),2))+D(I)
90 IF E(I)=24 THEN E(I)=0
100 IF E(I)<0 THEN E(I)=24+E(I)
110 T$(I)=STR$(E(I))+RIGHT$(T$(1),4)
120 IF LEN(T$(I))=5 THEN T$(I)="0"+T$(I)
130 PRINT C$(I);TAB(15);LEFT$(T$(I),2),
140 PRINT":";MID$(T$(I),3,2);":";RIGHT$(
T$(I),2);
```

```
150 NEXT I:GOTO 60
160 DATA NEW YORK, 0, MOSCOW, 8, RIO DE JANE
IRO, 2
170 DATA SYDNEY, 15, HONOLULU, -5, LONDON, 5,
CAIRO, 7
180 DATA TOKYO, 14, SAN FRANCISCO, -3, PARIS
 6
```

注：

TI$变量不能用INPUT语句来记入，此外，时间从23：59：59进到00：00：00后，时间00：00：01是不显示的。

2·3、8．6 CURSOR（光标）……（缩写格式：CU.）

格式：

CURSOR X．Y

X……座标（0到39）

Y……座标（0到24）

功能：

使用这个命令可以将显示屏上的光标移动到任一指定位置，而且可以和PRINT与INPUT语句一块使用，将字符指定到屏幕的任一位置上显示。

屏幕坐标是这样规定的，列的编排，从左到右。起始列为0，右末列是39，行的编排，从上到下，顶始为0，底末行是24，所以允许光标从上顶（00）到底部（39、24）之间的任一位置上移动。

例：

下列程序光标用（＊）号代替。（当然要像移动光标一样按键）。

```
10 X=0:Y=0
15 PRINT"🄲"
20 CURSOR X,Y:PRINT" *  ;
30 GET A$:IF A$="" THEN 30
40 CURSOR X,Y:PRINT"  ";
50 IF A$="↑"  THEN Y=Y-1  :REM  "UP   "
60 IF A$="↓"  THEN Y=Y+1  :REM  "DOWN
70 IF A$="←"  THEN X=X-1  :REM  "LEFT
80 IF A$="→"  THEN X=X+1  :REM  "RIGHT
90 IF X<0      THEN X=0
100 IF Y<0     THEN Y=0
```

```
110 IF X>38  THEN X=38
120 IF Y>24  THEN Y=24
150 GOTO 20
```

注意：

当X或Y的值不是整数时，在光标移动之前，计算机会自动取整。

移动光标的另一种方法是使用包含有TAB或SPC函数的PRINT语句和SPC函数（关于SPC函数的说明见后）

2·3、8．7 TAB

格式：

TAB（X）

X……数值表达式

功能：

TAB函数与PRINT语句一块使用，使光标移到从左边算起的X＋1个字符位置（这称为空格列表）。

例：

PRINT TAB（5）；“XYZ”；TAB（10）；“ABC”

0 1 2 3 4 5 6 7 8 9 0 1 2 ⇦实际上不显示

␣␣␣␣␣XYZ␣␣ABC

注：

列表仅仅是把光标向右移，因此，这个函数与PRINT语句一块使用时，如果光标已经移到在X指定的位置的右边，就什么也没有发生。

例：

PRINT TAB（5）；“XYZ”；TAB（5）；“ABC”

01234567890

␣␣␣␣␣XYZABC

2·3、8．8 SPC

格式：

SPC（n）

n……数值表达式

功能：

这个函数与PRINT语句一块使用，它的作用是输出n个空格串，这样也就把光标从

当前位置向右移动n个字符位。

例:

```
PRINT  SPC (5) . 'ABC'

01234567
␣␣␣␣␣ABC
```

2．下面的例子是说明TAB和SPC函数之间区别的

```
10  ?  TAB (2) : "ABC" ; TAB (6) : "DEF"
20  ?  SPC (2) : 'ABC" ; SPC (6) : "DEF"

01234567890123
␣␣ABC␣DEF␣␣␣␣␣␣
␣␣ABC␣␣␣␣␣␣␣DEF
```

SPC中的n是从当前位置起算，而TAB中的X总是从左边起始位起算的。

2·3、8．9 SET RESET

这些语句是用于接通（显示）或断开（关闭）在屏幕指定的光点位置。

格　式	功　能	X，Y座标限
SETX，Y<，C> X……规定X座标的数值表达式 Y……规定Y座标的数值表达式 C……颜色代码（0～7）	接通（显示）在屏幕上座标为X，Y的光点 （SET）	0≦X≦79 0≦Y≦49
RESET X，Y X…… Y…… 含义同上	关闭在屏幕上座标为X，Y的光点 （RESET）	0≦X≦79 0≦Y≦49

通过选择SET语句的不同代码，可以指定颜色，其代码如下：

（0）黑色
（1）蓝色
（2）红色
（3）紫色
（4）绿色
（5）淡蓝色
（6）黄色
（7）白色

通过SET语句可以同时显示4个光点，但是其中任一光点颜色改变，便会使其他点颜色也改变。

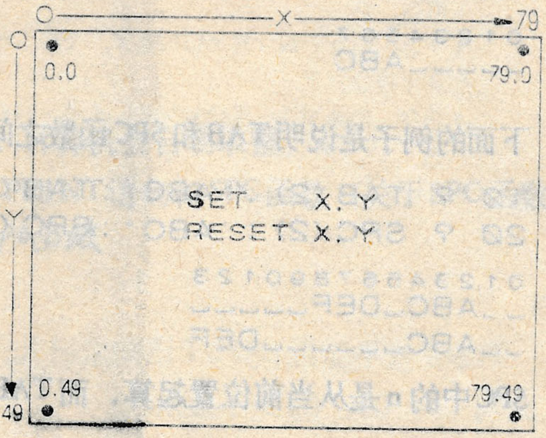
SET和RESET语句可以用来产生一些有趣形态，下面介绍一些例子：

1：在屏幕上显示一个光点

```
10  PRINT"C"
20  X=79:Y=49
30  SET  X, Y, 2
40  RESET  X, Y
50  GOTO  30
```

Turns dots on.

Turns dots off.



（2）把整个屏幕置为白色

```
10  PRINT"C"
20  FOR  X=0  TO  79
30  FOR  Y=0  TO  49
40  SET  X, Y, 7
50  NEXT  Y, X
60  GOTO  10
```

3、在屏幕的显示范围画出一个矩形框。

```
10  PRINT"C"
20  FOR  X=0  TO  79
30  SET  X, 0
40  SET  X, 49
50  NEXT  X
60  FOR  Y=0  TO  49
70  SET  0, Y
80  SET  79, Y
90  NEXT  Y
100  GOTO  100
```

4、这个程序是模仿一个小石子掷在水圹中泛起的波纹

```
10  X=40:Y=25
20  DEF  FNY(Z)=SQR(R*R-Z*Z)
30  PRINT C :SET  X, Y
40  R=R+5
50  FOR  Z=0  TO  R
60  T=FNY(Z)
```

```
70 SET X+Z, Y+T
80 SET X+Z, Y-T
90 SET X-Z, Y+T
100 SET X-Z, Y-T
110 NEXT Z
120 IF R<>25 THEN 40
130 GOTO. 130
```

5、这个程序是模仿一个球体在四壁之间弹跳

```
10 PRINT [C]"
20 FOR X=0 TO 79
30 SET X, 0:SET X, 49
40 NEXT X
50 FOR Y=0 TO 49
60 SET 0, Y:SET 79, Y
70 NEXT Y
80 X=79*RND(1) :Y=49*RND(1)
90 A=1:B=1
100 SET X, Y
110 IF X<2 GOSUB 200
120 IF X>78 GOSUB 200
130 IF Y<2 GOSUB 250
140 IF Y>48 GOSUB 250
150 RESET X, Y
160 X=X+A:Y=Y+B:GOTO 100
200 A=-A:MUSIC +A0" : RETURN
250 B=-B:MUSIC"A0": RETURN
```

2·3、9　音乐控制语句

这一节讨论MUSIC和TEMPO语句，这些语句的作用是由计算机控制演奏音乐，正如名字所暗示的，TEMPO（拍子）语句是规定演奏音乐的速度 音符(包括半音符以及高阶和低阶音阶）延续时间是由MUSIC音乐语句控制的。

TEMPO（拍子）：是在TEMPO语句说明中用一个数值变量或常量说明的，取值范围1（低）到7（高）

MEIODY（旋律）：是在MUSIC语句中用一个由音符集组成的字串来说明的。

音符说明：音阶　#高半音　音符名　拍数

2·3、9．1　MUSIC（缩写格式：MU．）

格式：MUSIC X$

X$ ……字串数据

自动演奏音乐

说明：

这个语句通过把字串或字串变量的值送到扬声器，计算机就自动产生旋律音响输出。演奏曲子速度是用TEMPO语句控制的。

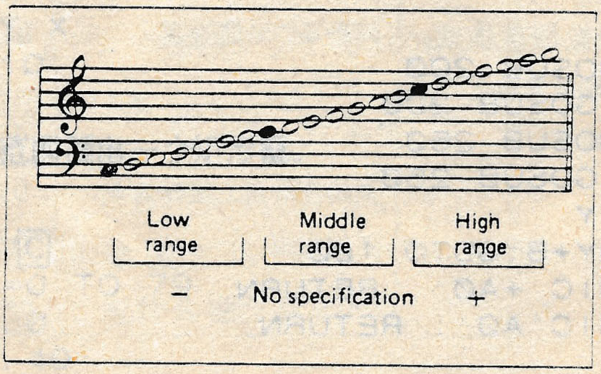
每个音符说明格式如下：

＜音阶说明＞＜ # 升高半音＞音符名＜拍数＞

使用“+”或“－”号说明音阶，如果二者都不用就是中音段。计算机可以输出如下图所示的三个声响范围。

例如，由黑园点下面指出C音符（8个音符的C调）的不同音阶。

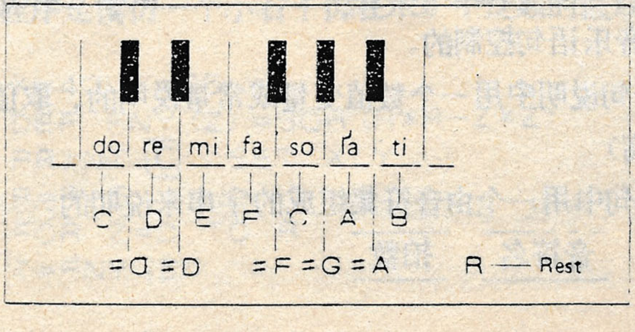
低 C .... -C
中 C...... C
高 C .... +C


Low range
Middle range
High range
－
No specification
＋

用以说明每个范围的音符说明符号如下：

CDEFGAB # R

下图所示是8个音调之间（do，re，mi ，fa，so，la，ti，do）关系和符号。高半音符号（#）用来规定半音符。休止间隔用“R”来说明。


do
re
mi
fa
so
la
ti
C
D
E
F
G
A
B
#C
#D
#F
#G
#A
R — Rest

```
30 M2$="A3+D2R0+D1+E2+D+#C3B+#C7+#C3"
40 M3$="A3+#C2R0+#C1+D2+#CB3A+D7+D3"
50 TEMPO 3
60 MUSIC MM$, M1$, M2$, M1$, M3$, M1$, M2$, M1
   $, M3$
70 END
```

2·3、10 数据文件的输入／输出命令

SAVE和LOAD命令是用来写读程序文本的，至于由程序处理的各类数据的写读则要使用其它命令，现把这些命令讨论如下：

格式		功能
WOPEN (缩写W.)	WOPEN（文件名）	在写数据于磁带前，打开数据文件，也可给数据文件命名。
PRINT／T (缩写？／T)	PRINT／Td_1，（d_2，d_3……dn） dn……数值数据或字串数据	把数据写入磁带，格式与PRINT语句相同。
ROPEN (缩写RO.)	ROPEN（文件名）	寻找在磁带上指定的数据名，并打开该文件，等待读入数据。
INPUT／T (缩写I.／T)	INPUT／T v_1，（v_2，v_3……vn） vn……数值数据或字串数据	读入磁带上指定的数据，方法类似INPUT语句从文件键盘输数。
CLOSE (缩写CLO.)	CLOSE	关闭已读完的磁带数据文件。

执行WOPEN和ROPEN语句时，在屏幕上无任何信息显示，而执行SAVE和LOAD命令则有显示。

注意：

当使用普通盒式录音机录制时，即使是在使用SAVE和LOAD命令去贮存和读程序时，没有碰到录制困难，但也不能用来录制数据文件。

例1：

下面的程序是把数字1～99写入盒式磁带

```
10 WOPEN "DATA"
20 FOR X=1 TO 99
30 PRINT／T X
40 NEXT X
50 CLOSE
```

```
60END
```

例 2：

下面程序是把例 1 准备的数据资料读出来。在执行这个程序之前，记住把带重绕回来。

```
10ROPEN “DATA”
20FOR X= 1 TO 99
30INPUT/ T A
40PRINT A
50NEXT X
60CLOSE
70END
```

例 3：

下面程序是建立一个由字串资料组成的数据文件。

```
10  DIM N$(5)
20  N$(1)="BACH"
30  N$(2)="MOZART"
40  N$(3)="BEETHOVEN"
50  N$(4)="CHOPIN"
60  N$(5)="BRAHMS"
70  WOPEN"GREAT MUSICIAN"
80  FOR J=1 TO 5
90  PRINT/T N$(J)
100  NEXT J
110  CLOSE
120  END
```

例 4：

下面程序是从例 3 建立的文件中读出字串资料，在执行这个程序之前要把盒式磁带反绕回来。

```
200  DIM M$(5)
210  ROPEN "GREAT MUSICIAN"
220  FOR K=1 TO 5
230  INPUT/T M$(K)
240  PRINT M$(K)
250  NEXT K
260  CLOSE
270  END
```

这些命令也可以用来建立包含有数字数据和字串数据的数据文件，然而如果在 INPUT/

T语句中的变量类型与数据类型不匹配，将会出现错误。所以一般来说，在一个数据文件中最好只用一种数据类型。

注意：

当用WOPEN语句打开一个顺序文件时，“文件名”可以省略，然而当在同一条磁带中有许多个文件时，可能会读错，因此，对于顺序数据文件，还是养成分配文件名的习惯为好。

下面程序是把每个学生的英语、法语、自然科学与数学的成绩录制成一个顺序的数据文件。

```
10 INPUT"ENTER NO.  OF  STUDENTS";N
20 DIM N$(N),K(N),E(N)
30 DIM R(N),S(N)
40 A$="GRADE IS"
50 FOR X=1 TO N
60 PRINT:PRINT "STUDENT NO.  ";X
70 INPUT"ENTER STUDENT NAME:  ";N$(X)     输入编号为X的学生姓名
80 PRINT "ENG  ";A$;:INPUT K(X)           输入英语成绩
90 PRINT "FREN";A$;:INPUT E(X)            输入法语成绩
100 PRINT. "SCI  ";A$;:INPUT R(X)         输入自然科学成绩
110 PRINT "MATH";A$;:INPUT S(X)           输入数学成绩
120 NEXT X
130 WOPEN"GRADES"          为了输入到盒式磁带打开文件名为“GRADES”数据文件
140 PRINT/T N              把这个班的学生数写入文件
150 FOR X=1 TO N
160 PRINT/T N$(X),K(X),E(X),R(X),S(X) 把姓名各科成绩写入文件
170 NEXT X
180 CLOSE              关闭盒式磁带文件
190 END
```

下面程序是先从盒式磁带中读出由上面程序建立的学生成绩数据文件，然后计算机显示每个学生平均成绩及这个班的各科平均成绩。

```
10 ROPEN "GRADES"             ←为了输入计算机打开盒式磁带中的名字为“GRADES”的数据文件
20 INPUT/T N                        读出这个班级的学生数
30 DIM N$(N),K(N),E(N)              定义各数组大小
40 DIM R(N),S(N)
50 FOR X=1 TO N
60 INPUT/T N$(X),K(X)                    读出学生娃名英语成绩
70 INPUT/T E(X),R(X),S(X)                读出法语、自然科学、数学成绩
80 NEXT X
```

```
90 CLOSE                                              关闭文件
100 PRINT TAB(10); "ENG ";
110 PRINT TAB(15); "FREN";
120 PRINT TAB(20); "SOI ";
130 PRINT TAB(25); "MATH"
140 FOR X=1 TO N
150 PRINT N$(X);TAB(10);K(X);
160 PRINT TAB(15);E(X);
170 PRINT TAB(20);R(X);
180 PRINT TAB(25);S(X);
190 PRINT TAB(30);(K(X)+E(X)+R(X)+S(X))/4
200 K(0)=K(0)+K(X):E(0)=E(0)+E(X)
210 R(0)=R(0)+R(X):S(0)=S(0)+S(X)
220 NEXT X
230 PRINT TAB(10);K(0)/N;TAB(15);E(0)/N;
240 PRINT TAB(20);R(0)/N;TAB(25);S(0)/N
250 END
```

第四节　内部函数

函数	符号	例子	说　　明
绝对值函数	ABS（X）	A＝ABS（X）	取｜X｜值赋给变量A 例：A＝ABS（2.9）→A＝2.9 A＝ABS（－5.5）→A＝5.5
符号函数	SGN（X）	A＝SGN（X）	取变量X的符号数赋给A，如果X为负把－1赋给A，如果X值为0，就把0赋给A，如果X值为正，就把1赋给A $A=\begin{cases}1 & (X>0)\\ 0 & (X=0)\\ -1 & (X<0)\end{cases}$ 例：当A＝SGN（0.4）执行后把1赋给变量A。
取整函数	INT（X）	A＝INT（X）	把最大的整数值赋给A，A的值小于等于变量X的值。 例：A＝INT（3.87）→A＝3 A＝INT（0.6）→A＝0 A＝INT（－3.87）→A＝－4
三角函数	SIN（X）	A＝SIN（X） A＝SIN（30＊PAI（1）/180）	把X的正弦值（当X为弧度时）赋给变量A，如果X是角度那么使用这个函数求正弦值之前，必须转换成弧度。由于1度等于π/180弧度，所以弧度数由度数乘上PAI（1）/180取得。例如：30°等于＊PAI（1）/180，弧度，对于COS，TAN和ATN同样适用

三角函数	COS（X）	A＝COS（X） A＝COS（200＊PAI（1）／180）	取X的余弦值赋给变量A
	TAN（X）	A＝TAN（X） A＝TAN（Y＊PAI（1）／180）	取X的正切值赋给变量A
	ATN（X）	A＝ATN（X） A＝180／PAI（1）＊ATN（X）	取X弧度的反正切赋给变量A，这个值的范围在：－PI／2～PI／2之间.
平方根函数	SQR（X）	A＝SQR（X）	计算X的平方根并把结果赋给变量A,X必须是正数或0.
指数函数	EXP（X）	A＝EXP（X）	计算e^x的值，并把结果赋给变量A，
常用对数函数	LOG（X）	A＝LOG（X）	计算X的常用对数，并把结果赋给变量A，
自然对数函数	LN（X）	A＝LN（X）	计算X的自然对数（$\log e^x$）并把结果赋给变量A，
园周率	PAI（X）	A＝PAI（X）	把X乘PI的结果赋给变量A，
弧度	RAD（X）	A＝RAD（X）	把X值（X是角度）转换成弧度数，然后赋给A，

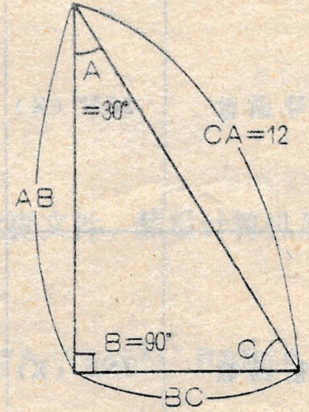
应用内部函数的例子：

例1：

用BASIC程序求解三角形的各元素

在右图中三角形的A角为30°，B角是直角斜边长是12，下面程序是求三角形的另一角C和其它边长以及三角形面积。

```
10 A=30:B=90:CA=12
20 AB=CA*COS(A*PAI(1)/180)
30 BC=CA*SIN(A*PAI(1)/180)
40 S=AB*BC/2
50 C=180-A-B
60 PRINT "AB=";AB, "BC=";BC, "CA=";CA
70 PRINT "AREAS=";S
80 PRINT "A=";A, "B=";B, "C=";C
90 END
```



例2：

现让我们把程序50改用ATN，通过求（AB／BC）的反正切函数来求C角

```
60 IF R=2 THEN N2=N2+1
70 IF R=3 THEN N3=N3+1
80 IF R=4 THEN.N4=N4+1
90 IF R=5 THEN N5=N5+1
100 IF R=6 THEN N6=N6+1
110 NEXT J
120 P1=N1/N:P2=N2/N:P3=N3/N
130 P4=N4/N:P5=N5/N:P6=N6/N
140 PRINT P1,P2,P3,P4,P5,P6
150 END
```

上面结果怎样呢？如果骰子投掷次数足够大，那么每个面朝上次数所占的百分数大体上是相当接近或相同的。从数学上说，投掷六次平均每面向上该出现一次，或者说占百分数大约是 16.7%．随着投掷次数的增加，各面朝上所占的百分数会越来越接近 16.7%。

让我们在程序中利用RND函数，测验你解算随机三角形面积的能力，在这里 RND函数是用来确定三角形的三个边的边长的，然后你自己计算三角形面积，并把答案送入计算机核对。

```
10 DIM A(3),L$(4)
20 FOR J=1 TO 4
30 READ L$(J):NEXT J
40 FOR J=1 TO 3
50 A(J)=INT(20*RND(1))+1
60 NEXT J
70 IF A(1)>=A(2)+A(3) GOTO 40
80 IF A(2)>=A(1)+A(3) GOTO 40
90 IF A(3)>=A(1)+A(2) GOTO 40
100 W=(A(1)+A(2)+A(3))/2
110 T=W :FOR J=1 TO 3
120 T=T*(W-A(J)):NEXT J
130 SS=SQR(T):S=INT(SS)
140 IF SS-S>0.5 THEN S=S+1
150 PRINT   ■■■■■
160 PRINT          SOLVE FOR THE AREA OF THE
    FOLLOWING TRIANGLE"
170 PRINT          ROUND YOUR ANSWER TO THE
    NEAREST WHOLE NUMBER"
180 PRINT
190 PRINT TAB(8): A
200 PRINT TAB(8): ⧄⧅"          ;TAB(15);L$(1)
    ;A(1)
210 PRINT TAB(7): ⧄  ⧅        ;TAB(15);L$(2)
    ;A(2)
220 PRINT TAB(6):,⧄    ⧅      ;TAB(15);L$(3)
     ;A(3)
230 PRINT TAB(5): ⧄      ⧅
```

```
240 PRINT TAB(3): B0          ⊠C.
250 PRINT TAB(4): □□□□□□□□□□□
260 PRINT  ■■■"
270 PRINT TAB(3);L$(4);
280 INPUT Y
290 IF Y<S THEN PRINT "      OK!! ":GOTO
    40
300 IF Y<S THEN PRINT " TOO SMALL!"
    :GOTO 320
310 PRINT "    TOO LARGE!"
320 PRINT "■■";
330 PRINT TAB(24);SPC(25):PRINT ■ ;
340 GOTO 270
350 DATA LENGTH SIDE AB= LENGTH SIDE BC=
360 DATA LENGTH SIDE CA=, AREAS OF TRIAN
    GLE ABC IS
```

注意：

如果记入的X值小于等于0时，对于一个给定的X值，结果总是相同的，这是因为记入0或一个负数，使伪随机数发生器重新恢复到随机数列的第一个数（地点）。

第五节　字串函数

2·5、1　LEN

格式：

LEN（X$）

X$……字串表达式

功能：

这个函数得到字串X$的字符数，这个数包括在屏幕上不显示的空格和该字串的控制字符以及字母、数字和符号。

例（1）：

10A$="ABCDEFG"

20PRINT LEN（A$）.

```
RUN
 7
```

（2）

以下程序是利用LEN函数在屏幕上画矩形：

```
10 ? ■":?"ENTER 3OR MORE ASTERISKS"
20 INPUT A$
30 FOR I=1 TO LEN(A$)-2
40 PRINT TAB(2):"*":SPC(LEN(A$)-2):"*"
50 NEXT I
60 PRINT TAB(2):A$:GOTO 20
```

（3）

如下所示是利用LEN函数产生字符阵列：

```
10 S$= SHARP BASIC"
20 FOR I=1 TO LEN(S$)
30 ? RIGHT$(S$. I)
40 NEXT I
50 END
RUN
C
IC
SIC
⋮
SHARP BASIC
```

(Example 4)

```
PRINT LEN(STR$(PAI(1)))↲
 9
```

PAI（1）—得到园周长与其直径的比，是一个含有8位数的常数3.1415927（PI的近似值）用LEN函数来计算由STR$函数把这个常数转换成字符串的字符数，得到总的字符数是9。

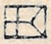
2·5、2 LEFT$，MID$，和RIGHT$函数

LEFT$，MID$和RIGHT$函数是用来从字串表达式的左端，右端，中间截取字符。

格式 X$：字串表达式 m和n：数值表达式	功　　能	例　　子	附　注
LEFT$（X$，n）	得到从字串表达式X$的左边算起n个字符组成的字串。	B$＝LEFT$（A$，2） B$← $\boxed{AB}$ CDEFG 从字串变量A$的左边起取二个字符赋予字串变量B$，所以B$＝“AB”	0≦n≦255
MID$（X$，m，n）	得到从字串表达式X$的第m个字符起，取n个字符组成新的字符串。	B$＝MID$（A$，3，3） B$← AB $\boxed{CDE}$ FG 从字串变量A$的第三个字符起取三个字符代入字串变量B$	1≦m≦255 0≦n≦255
RIGHT$（X$，n）	得到从字符串变量A$的右端开始取n个字符组成的字串。	B$＝RIGHT$（A$，2） B$← ABCDE $\boxed{FG}$ 从字符串变量A$的右端开始取二个字符赋给字串变量B$，所以B$＝“FG”	0≦n≦255

2·5、3

ASC和CHR$函数（ASC美国信息交换标准代码）

格式	功　　能	例　子
ASC（X$） 字串表达式	得到字串表达式的第一个字符的ASCII代码	X＝ASC（“A”） 把65（字母A的ASCII代码）代入变量X Y＝ASC（“$\boxed{S}$ HARP”） 把83（是字串的第一个字母S的ASCII代码）代入变量X
CHR$（X） X：算术表达式	得到对应于X算术表达式的值的全部ASCII代码的字母（如果X的值小于33，就得不到字符，因此，如果希望得到空格应该用PRINT“ — ”或SPC（1）而不应用CHR$（32）	A$＝CHR$（65） 把对应于ASC代码65的字母A赋予字串A$，这个函数可以用来显示字母，但不能从键盘输入如： PRINT CHR$（107）、将显示绘图符号

注意：

ASCII 代码是计算机常用的标准代码系统，这些代码是用 8 位数来表示字母表中的字母、数字以及象美元号、问号等这样的符号。

2·5、4　VAL和STR$函数

格式	功能	例子
STR$（X） X：表示算术表达式	把算术表达式X的值转成字符串	A$= STR$（−12） 把字符串“−12”赋给字串变量A$ B$= STR$（70 * 33） 把字符串“2310”赋给字串变量B$ C$= STR$（1200000 * 5000） 把字符串“6E+09 ”赋给字串变量C$。 注：显示正数时首部带有一个空格，其正号被忽略然而通过用STR$函数转换成字符串后并不包含这个空格。例如 10 A= 123　　40 PRINT B$ 20 B$= STRS（A）　　RUN 30 PRINT A　　− 123 123
VAL（X$） X$：字串表达式	把一个数字组成的字串转换成数值，这个函数是STR$的反函数	A= VAL（“123”） 把字串“123”转成数值123，并赋给数值变量A。

以下说明应用上面讨论的函数以表格型式　（以小数点为基准）把数值显示出来，例如下面的数字

1.23456
12.3456
10
1
1234

用PRINT语句显示从DATA语句中读出的数值如果希望小数点固定在一列上可用如下程序：

```
10 FOR X=1 TO 5
20 READ A
30 L=6-LEN(STR$(INT(A)))
40 PRINT TAB(L);A
50 NEXT:END
60 DATA 1.23456,12.3456
70 DATA 123.456,1234.56
80 DATA 12345.6
```

```
    1.23456
   12.3456
  123.456
 1234.56
12345.6
```

第六节　彩色显示语句

MZ—700的最大特点之一，是允许用 8种颜色中的任何颜色显示字符和图像。

2·6、1　颜色（COLOR）语句……（缩写：COL.）

格式：

COLOR X，Y，C＜，b＞

X……X座标（0～39）

Y……Y座标（0～24）

C……字符颜色（0～7）

b……背景颜色（0～7）

功能：

这个语句用来设置在屏幕指定位置上的字符（前景）和背景颜色，为了显示字符（前景）和背景，可以使用如下所示的 8种颜色中的任何颜色。

颜色号	颜色
0	黑
1	兰
2	红
3	紫
4	绿
5	青兰
6	黄
7	白

例：

（1）改变全屏的背景颜色

COLOR，，，2……（把显示字符的背景颜色改为红色）。

（2）改变全屏的前景颜色（对所有字符均用此色显示）。

COLOR，，3……（一切显示符的颜色改为紫色）。

（3）改变全屏的前景和背景颜色

COLOR，，1，0……（把显示字符的颜色改为兰色而显示字符的背景颜色，改为黑色）。

（4）改变屏幕指定位置上的背景颜色

COLOR 2，2，，4……（把屏幕2，2座标点的颜色改为绿色）。

（5）改变屏幕指定位置上的前景颜色

COLOR 3，2，7……（把座标3，2点上的前景颜色改为白色）。

（6）改变屏幕指定位置上的前景和背景颜色

COLOR 4，2，4，2……（把座标为4，2的位置上的前景颜色改为绿色，背景改为红色）。

2·6、2 对打印（PRINT）语句加入颜色说明

格式：

```
┌ PRINT ┐ 〔f,b〕 ┌ 变量  ┐< ┌ ; ┐ ┌ 变量  ┐ ┌ ; ┐      …… > ：
┤       ├        ┤ 常数  ├  ┤   ├ ┤ 常数  ├ ┤   ├
└   ?   ┘        └ 表达式┘  └ , ┘ └ 表达式┘ └ , ┘
```

或：

```
┌ PRINT ┐ 〔f,b〕 USING "格式字串"；变量< ┌ ; ┐ 变量……>
┤       ├                                └ , ┘
└   ?   ┘
```

f ……说明前景（字符）颜色（是一个0～7的数）

b ……说明背景颜色（是一个0～7的数）

功能：

把颜色说明加入在21页中讨论的PRIKT和PRINT USING语句，使其可以用各种颜色显示字符。在上面的格式中，f是指前景（字符）颜色，而b是指背景颜色，如果只说明字符颜色，则当前的背景颜色不变，这样做的话，在说明字符颜色之后要加写入一个逗号（，）。

如果仅仅规定背景颜色，则显示字符的当前颜色不变，在这种情况下逗号必须放在背景颜色说明之前。

例如：（1）

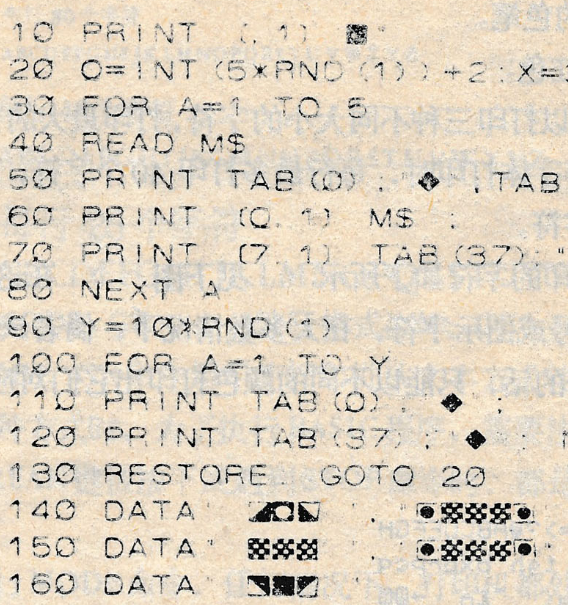
PRINT〔6，5〕"ABCDE"……显示字符"ABCDE"时，字符颜色是黄色，背

景颜色是青兰色。

PRINT〔，4〕"FGHIJ"………显示字符"FGHIJ"时，字符颜色仍是黄色，背景则改用绿色。

PRINT〔7，〕"VWXYZ"……显示字符"VWXYZ"时，字符颜色为白色，而背景颜色仍是绿色。

（2）让我们试试把颜色加入在30页所示的汽车竞赛程序中去，看显示结果如何。



```
10 PRINT (,1) ▩"
20 O=INT(5*RND(1))+2:X=33*RND(1)
30 FOR A=1 TO 5
40 READ M$
50 PRINT TAB(O);"◆";TAB(X);
60 PRINT (O,1) M$;
70 PRINT (7,1) TAB(37);"◆"
80 NEXT A
90 Y=10*RND(1)
100 FOR A=1 TO Y
110 PRINT TAB(O);"◆";
120 PRINT TAB(37);"◆":NEXT
130 RESTORE:GOTO 20
140 DATA"◢◘◣","◙▩▩▩◙"
150 DATA"▩▩▩","◙▩▩▩◙"
160 DATA"◥■◤"
```

用普通的打印（PRINT）语句（没有规定颜色）显示字符时，前景颜色和背景颜色是用最近的COLOR语句规定的颜色显示。

第七节　彩色绘图打印命令

现将MZ—731或MZ—721，MZ—711与MZ1PO1颜色绘图打印机连接时，使用的彩色绘图打印命令讨论如下，彩色绘图打印机可以使用二种方式中的任一种方式，文本方式，（程序列表，计算结果或其他字符数据输出）或绘图方式（画曲线和画画）。

此外，四种颜色（黑色、兰色、绿色、红色）中的任一种颜色均可用于字符和绘图打印输出，这种性能在绘图方式中特别有用。

2·7、1　关于彩色绘图打印机的一般介绍

（1）彩色绘图打印机的操作可以使用如下二种方式之一，文本方式（输出计算结果，程序列表，和其他字符数据）或绘图方式（用画曲线和画图），在同一时间内打印机只能使用一种方式工作（当打印机在文本方式工作时，绘图打印命令会被忽略、反

之亦然。）

（2）打印机从绘图方式转换为文本方式时，打印机参数将被置 0（换句话说，笔的 X，Y 座标置位被清）。

（3）主机 MZ--731 的电源开关接通时，打印机开始运行，而不必另设电源开关。

（4）下面开关是用于控制打印机工作的。

a．进纸开关……把纸向前推出；

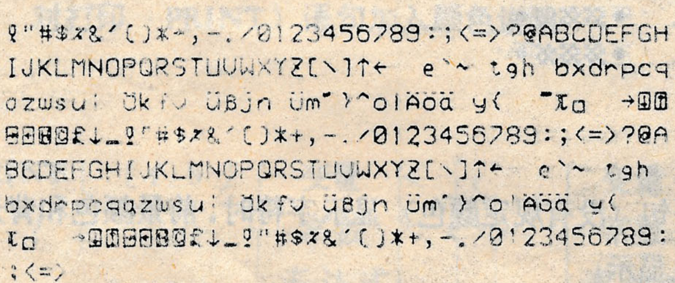
b．复位开关……恢复打印机初始状态；

c．换笔开关……用于换打印机上的色笔。

（5）有四支色笔，黑色、兰色、绿色、红色。

（6）当打印机在文本方式中使用时，可以打印三种不同大小的字符，打印最大的字体时，每行最多打印 26 个字符，使用中等字体打印时，每行最多打印 40 个字符，而用最小字体打印时每行最多打印 80 个字符。

当在文本方式中使用打印机时，可打印的字符如下所示（　见下图　）在这种方式中打印机的输出，再没有另外的字母，符号或图示字符，在大多数情况下，倘若试图用 PRINT／P 或 LIST／P 命令打印图示字符的话，只能以不同的颜色打印出它们对应的 16 进制的 ASCLL 代码。



```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[\]↑←  e`~ tgh bxdrpcq
ozwsu: Ökfv üßjn Üm")^olÄöä y(  ‾£α  →▣▣
▣▣▣▣£↓_!"#$%&'()*+,-./0123456789:;<=>?@A
BCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←  e`~ tgh
bxdrpcqqzwsu: Ökfv üßjn üm')^o'Äöä y(
£α  →▣▣▣▣▣▣£↓_!"#$%&'()*+,-./0123456789:
;<=>
```

2·7、2　打印机初始状态设置

当 BASIC 解释程序 1Z—013B 启动之后，打印机的状态设置如下：

（1）笔色：黑色

（2）笔位：车架的左边（一页的第一行）

（3）方式：文本方式

（4）打印字符尺寸：每行 40 个字符……（标准尺寸）每页 66 行。

2·7、3　方式说明命令

这些命令使打印机置于文本方式，以打印输出字母和数字，主机电源接通以后，文本方式生效，字符大小是每行 40 个字符（40／行）。

（1）MODE TN……（缩写格式：M．TN）

这个命令是使打印机从绘图方式返回文本方式，且置字符大小是每行 40 个字符。

（2）MODE TL……（缩写格式：M．TL）

这个命令是使打印机从绘图方式返回文本方式，且设置字符大小每行 26 个字符。

（3）MODE TS……（缩写格式：M．TS）

这个命令使打印机从绘图方式返回文本方式，且设置字符大小是每行 80 个字符，

```
        ***  CHARACTER  MODE  ***
每行 80 个字符
ABCDEFGHIJKLMNOPQRSTUVWXYZ
每行 40 个字符
ABCDEFGHIJKLMNOPQRSTUVWXYZ
每行 26 个字符
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

（4）MODE GR……（缩写格式：M．GR）

MODE GR命令用于使打印机从文本方式转换为绘图方式，以便打印输出曲线或图形。交换这种方式时，为了执行BASIC程序，需要注意变换方式前所使用的字符尺寸，这是因为当BREAK键被按下或遇到STOP命令时，都是返回正常文本方式。

注意：

执行MODE命令，任何情况下，打印机都处于初始状态，而不考虑当前色笔状况和字符大小。

2·7、4 色笔选择命令

```
PCOLOR n    ┌ n： 0  黑色
            │ n： 1  蓝色
            ┤ n： 2  绿色 ……（缩写格式：PC．）
            └ n： 3  红色
```

此命令用于规定打印字符和绘图时使用那种颜色，n是0～3的一个数，0对应于黑色，1对应于蓝色，2对应于绿色，3对应于红色。

在文本方式中使用PCOLOR命令时，不考虑初始状态笔色，执行指定数字n（0～3）所对应的颜色。

执行PRINT／P CHR$（29）命令时，在此命令后打印的字符颜色是此命令之前使用的打印颜色的下一种颜色，如当前是黑色，在此命令后打印的字符则是蓝色，当前是红色此命令后则用黑色打印。

在文本方式，绘图方式均可输入此命令。

2·7、5　文本方式命令

2·7、5．1

检查（TEST）命令……（缩写格式：TE.）

格式：

TEST

功能：

此命令使打印机打印出四种不同颜色的正方形。以核对颜色说明，检查笔的墨量等等。（只在文本方式中用）

□□□□

0 1 2 3……在"PCOLOR　n"中的n值

黑蓝绿红

2·7、5．2　移纸（SKIP）命令

格式：

SKIP　n

n是－20～20中的一个数

功能：

这个命令用于移动纸，当n的值为正时，此命令把纸向前移动n行，如果n值是负，则把纸向后缩回n行。

注意：

当打印机处于绘图方式，执行这个命令则产生"PRINTER　MODE　ERROR"（打印机方式错）错误。

2·7、5．3　页行（PAGE）命令

格式：

PAGE　n

n：是大于等于1，小于等于72的一个整数（$1 \leq n \leq 72$）

功能：

这个命令用于规定每页打印多少行（仅在文本方式中使用）

2·7、5．4　列表（LIST／P命令……（缩写格式：L／P）

格式：

LIST／P或LIST／P（Ls—Le）
Ls……起始行号
Le……最终行号

功能：

这个命令用于从打印机上输出内存中全部或部分程序行。关于使用说明和LIST一样，可参看LIST命令的说明，但应注意的一点是：当程序中包含有绘图字符时，它们中的大多数，是用不同颜色以16进制的ASCII代码形式打印出来。（关于ASCII代码见/22页）这个命令仅在文本方式中使用。

2·7、5．5　打印（PRINT／P）命令……（缩写格式：？／P）

格式：

PRINT／P＜I₁，d₁，I₂，d₂……In　dn＞
In……输出列表（算术或字串表达式）
dn……定义符

功能：

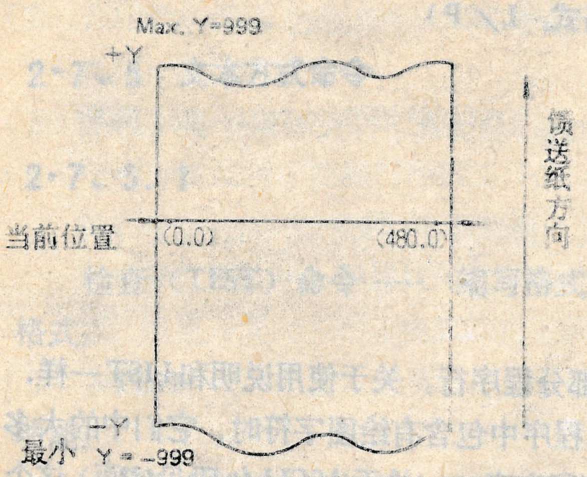
这个命令是把输出数送打印机列表输出。使用此命令的更详细说明参看 PRINT 命令，关于绘图字符的打印输出见本页。

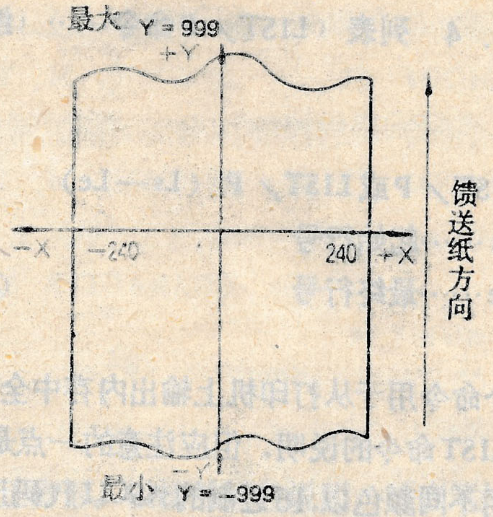
2·7、5．6　格式打印（PRINT／PUSING）命令……（缩写格式：？／PUSI.）

除输出是从打印机输出外，这个语句和在22页中讨论的PRINT　USING语句完全相同。

2·7、6　绘图方式语句

在MODE　GR语句执行之后，绘图方式语句才生效，当执行这个语句时，当前的笔所在位置为原点（X＝0，Y＝0）因此原点可以设在任何位置，要是不小心把说明的位置，设在打印区外时，会损坏色笔或产生其它问题。


Max. Y=999
+Y
当前位置
(0.0)
(480.0)
馈送纸方向
−Y
最小 Y=−999


最大 Y=999
+Y
−X
−240
240
+X
馈送纸方向
−Y
最小 Y=−999

在执行MODEGR后X～Y	原点被移到纸的中心后 X～Y
座标X范围是0～480	座标
Y范围−999～+999	（MOVE 240，−240：HSET）

2·7·6·1 画线（LINE）语句

格式：

LINE x_1，y_1 <，x_2，y_2，……，xi，yi >

或

LINE %n，x_1，y_1，<，x_2，y_2，……，xi，yi，>

n……是1～16的一个整数

xi……x座标的算术表达式（xi＝−480～480这个界限的改变取决于当前笔的位置）。

yi……y座标的算术表达式（yi＝−999～999）

功能：

这个语句是从笔当前位置到座标为X_1，Y_1位置画一条线，然后从X_1，Y_1到X_2，Y_2画一条线等等，n用来规定画线的型式如下所示：

n＝1　　实线

n＝2～16　虚线

如果%被省略，则设定为先前的n值，n的初始值为1（实线）例：

```
10 MODE GR        ……置绘图方式
20 LINE 240,0     ……从原点到纸中心画一条线
30 LINE 240,−240
```

```
40 LINE  0, -240
50 LINE  0,0      ……画一条线到原点
60 MODE  TN       ……返回文本方式
```

（2）下面的程序是画与上例相同的正方形。

```
10 MODE  GR
20 LINE  240, 0, 240, -240, 0. -240, 0, 0
30 MODE  TN
```

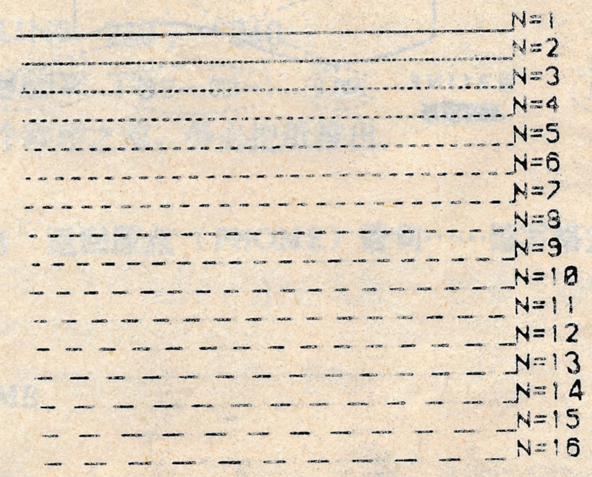
（3）下面程序是画一个边长为 240 个单位的三角形

底

```
10 MODE  GR
20 SQ=INT(120*SQR(3))
30 LINE %2.240.0.120.-SQ.0.0
40 MODE  TN
```

n 值与线的对应型式如下所示

```
*** LINE 1-16 ***
```


N=1
N=2
N=3
N=4
N=5
N=6
N=7
N=8
N=9
N=10
N=11
N=12
N=13
N=14
N=15
N=16

说明：

n = 1实线

n = 2点线

n = 3～16从点线复虚线

2·7、6. 2　画相对线段（RLINE）语句（缩写格式：RL.）

格式：

RLINE　x_1, y_1 <, x_2, y_2, ……xi, yi ……>

RLINE　% n, x_1, y_1, <, x_2, y_2, ……, xi, yi ……>

n……是 1～16的整数

xi……x 座标的数值（-480～480）

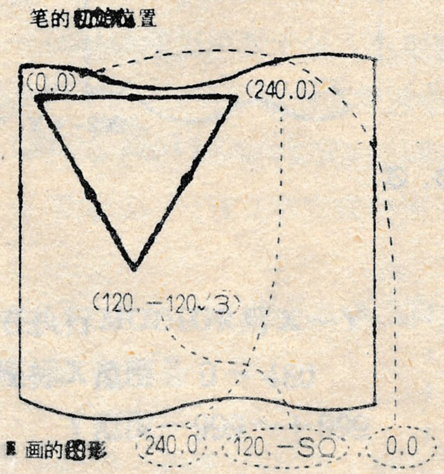
yi……y 座标的数值（-999～999）

功能：

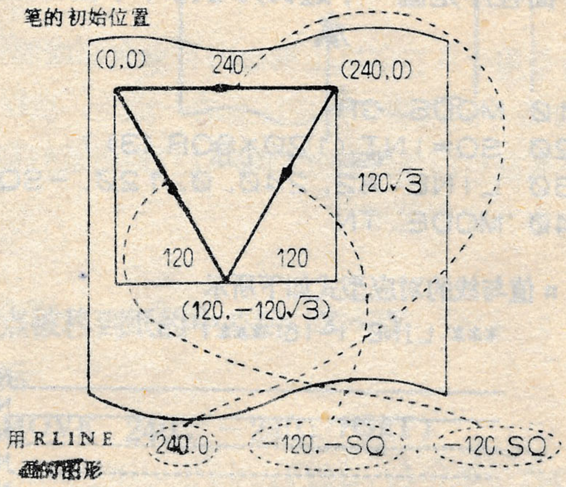
这个语句是从笔的当前位置到相对座标X_1，Y_1所指位置画一条线，然后又从那个点起（即X_1，Y_1）到相对座标X_2，Y_2所指位置画一条线等等，n的意义与LINE语句相同。

例：这个程序画与上面例 3 相同的三角形

```
10 MODE GR
20 SQ=INT(120*SQR(3))
30 RLINE %1. 240. 0. -120. -SQ. -120. SQ
40 MODE TN
```


笔的初始位置
(0.0)
(240.0)
(120.-120√3)
240.0
120.-SQ
0.0

用LINE画的图形


笔的初始位置
(0.0)
240
(240.0)
120√3
120
120
(120.-120√3)
240.0
-120.-SQ
-120.SQ

用RLINE画的图形

2·7、6. 3 移笔（MOVE）语句

格式： MOVE X，Y

x ……代表x座标的一个整数（－480～480）

y ……代表y座标的一个整数（－999～999）

功能：

这个语句的作用是先抬笔，以后把笔移到指定位置（X，Y）处。

例：这个程序是画一个边长为 480 个单位的“＋”字形。

```
10 MODE GR
20 LINE  480, 0
30 MOVE  240, 240
  在（480， 0）位置上把笔抬起然后移到240．240）
40 LINE  240, -240
50 MODE TN
```

2·7、6．4 相对移笔（RMOVE）语句……（缩写格式：RM）

格式：

RMOVE X、Y

x……代表 x 相对应座标一个整数（－480～480）

y……代表 y 相对应座标的一个整数（－999～999）

功能：

这个语句是抬笔，然后移到相对座标（△X，△Y）所指的位置上。

例：下面的程序是画一个与上面用MOVE语句的例子所画的相同的“＋”字形。

```
10 MODE GR
20 LINE 480，0
30 RMOVE －240，240……在（480，0）位置上抬笔，然后把笔沿X方向移
                        －240个单位，沿Y方向移 240个单位。
40 LINE 240，－240
50 MODE TN
```

执行这个程序之前，务必把纸推出。

2·7、6．5 返回原点（PHOME）语句……缩写格式：（PH．）

格式：

PHOME

功能：

这个语句使笔返回原点

例：

下面例子是画一个与上面用MOVE语句的例子相同的“＋”字形，然后用红色画个框住“＋”字形的正方形。

```
10 MODE GR
20 LINE ·480. 0 :MOVE 240. 240
30 LINE 240. -240
40 PHOME                        笔返回原点
50 PCOLOR 3
60 LINE 0. 240. 480. 240. 480. -240. 0. -240. 0.
0
70 MODE TN
```

2·7、6．6　设原点（HSET）语句（缩写格式：H．）

格式：

HSET

功能：

这个语句的功能是把当前笔的位置做为新的原点，绘图时，使用这个特点，可以把原点设置在最合适的位置上，为此，在执行这个命令之前，常常执行一个MOVE语句。

例：（画射线）

```
10 MODE GR
20 MOVE  240，－240
30 HSET……………………设置新原点
40 FOR I＝1 TO 360 STEP 30
50 LINE  240＊COS（PAI（1）＊I／180），240＊SIN(PAI（1）＊I／180）
60 PHOME
70 NEXT
80 MODE TN
```

2·7、6．7　绘图方式打印（GPRINT）语句……缩写格式：（GP．）

格式：

GPRINT 〔n，@〕，X$

GPRINT X$

n ……说明字符尺寸的一个整数（0～63）

@……说明字符打印方向的一个整数（@＝0～3）

X$……字符串

功能：

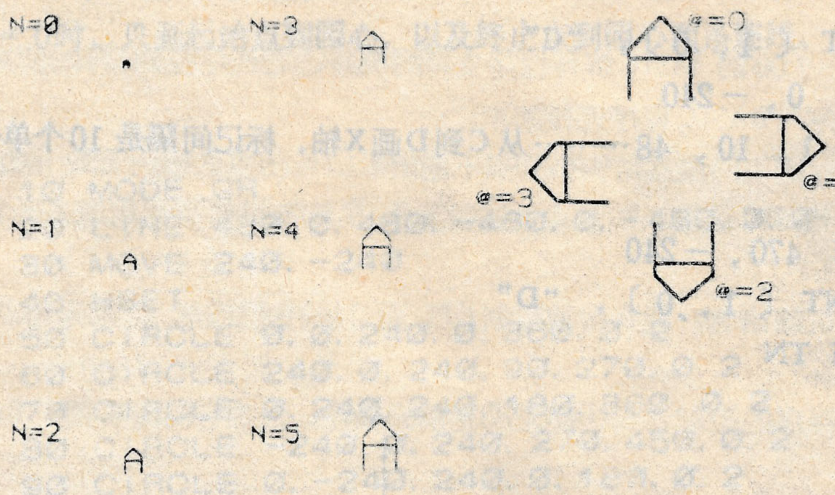
这个语句用规定的尺寸和方向打印字串（X$）的字符，当n＝0时，每行打印80个字符，当n＝1时，每行打印40个字符，当n＝2时，每行打印26个字符。

当n和@被省略时，取先前的设定，它们的初值是n＝1，@＝0。

例如：

```
10 MODE GR
20 GPRINT "A" ……用绘图方式打印 "A"
30 GPRINT 〔2，2〕，"A" ……以每行打印26个字符方式，打印倒写 "A"
                                  即（∀"）
```

下面表示出各种打印方式



2·7、6．8 画座标轴（AXIS）语句（缩写格式: AX．）

格式:

AXIS X，P，r

x……用于说明画座标轴的一个整数（0或1）

p……用于说明标度间隔的一个整数（－999～999）

r……用于规定打印标度个数的一个整数。

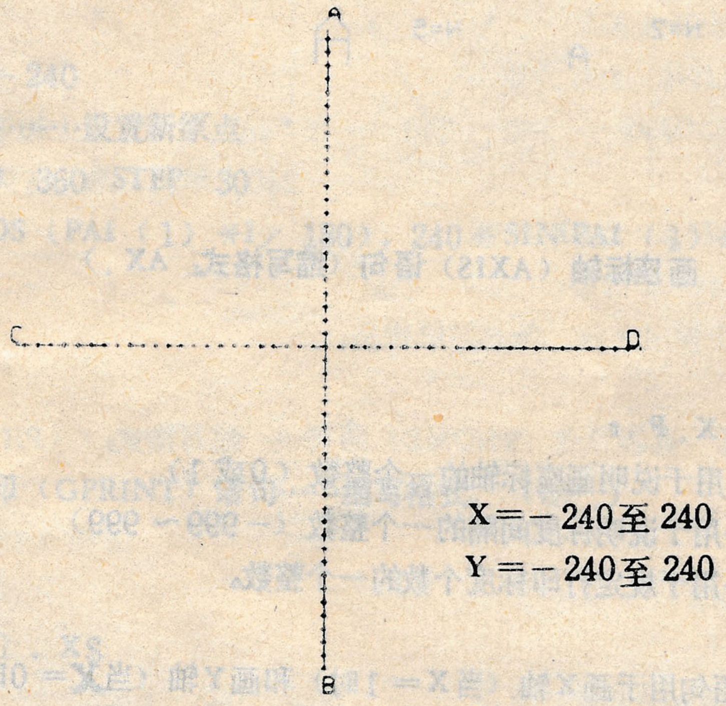
功能:

这个语句用于画X轴（当X＝1时）和画Y轴（当**X**＝0时），r是画标度的个数，P是标度的间隔。

例：下面的例子是画X和Y座标轴，标度标记从－240～240，标度间隔为10个单位。

```
10 MODE GR……………………置打印机绘图方式

20 MOVE 240，5
30 GPRINT 〔1，0〕，"A"
40 MOVE 240，0……………抬笔并移到A位置（240，0）

50 AXIS 0，－10，48………从A到B画Y轴，标度标记间隔10个单位

60 MOVE 240，－500
70 GPRINT 〔1，0〕，"B"
```

```
80 MOVE  0，－240………… 抬笔并移到位置C（0，－240）

90 GPRINT〔1，0〕，“C”
100 MOVE  0，－240
110 AXIS  1，10，48………从C到D画X轴，标记间隔是 10个单位

120 MOVE  470，－240
130 GPRINT〔1，0〕，“D”
140 MODE  TN
```



在设定X和Y轴交点作为新的原点后，画座标方法与普通的苗卡儿座标完全相同。

2·7、6．9 CIRCLE 缩写格式：（CI．）

格式：

CIRCLE x，y，r，s，e，d
x，y………园心位置（－999～999）
r…………半径
s…………起始角（度）
e…………终止角（度）
d…………步角（度）

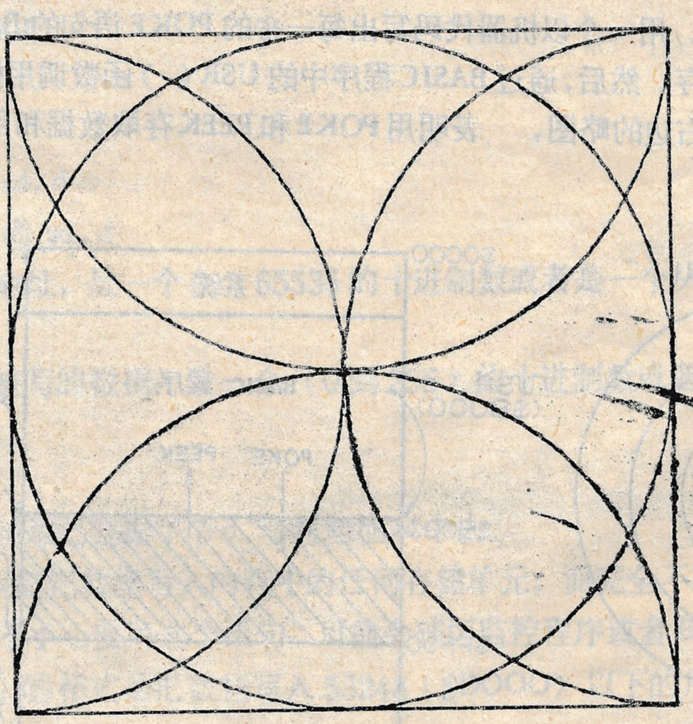
功能：

这个语句是以x，y为园心，以r为半径，以d为步长，画一个园弧或园，当s＝0，e＝360和d＝0.2时，是画一个完整的园。

实际上这个语句是画一个多边形，因此，为了使园形园滑，d必须尽可能地小，s必须小于e，当d＝0时，只画起始点到园心，以及终止点到园心两条连线。

例如：

```
10 MODE GR
20 LINE 480. 0. 480. -480. 0. -480. 0. 0
30 MOVE 240. -240
40 HSET
50 CIRCLE 0. 0. 240. 0. 360. 0. 2
60 CIRCLE 240. 0. 240. 90. 270. 0. 2
70 CIRCLE 0. 240. 240. 180. 360. 0. 2
80 CIRCLE -240. 0. 240. 270. 450. 0. 2
90 CIRCLE 0. -240. 240. 0. 180. 0. 2
100 MODF TN
```



第八节　机器语言程序的控制语句

本节讨论MZ—700的BASIC解释程序支持的机器语言程序控制语句．使用这些语句，可以把机器语言程序与BASIC程序连接起来，

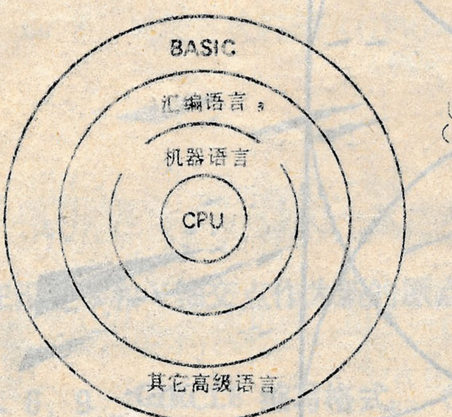
计算机程序设计语言的层次结构格式如下图所示。当使用低级语言时（如汇编语言），要求高级语言（如BASIC语言）能自动地进行工作。虽然高级语言使用起来既方便又容易，但它不能直接控制CPU。

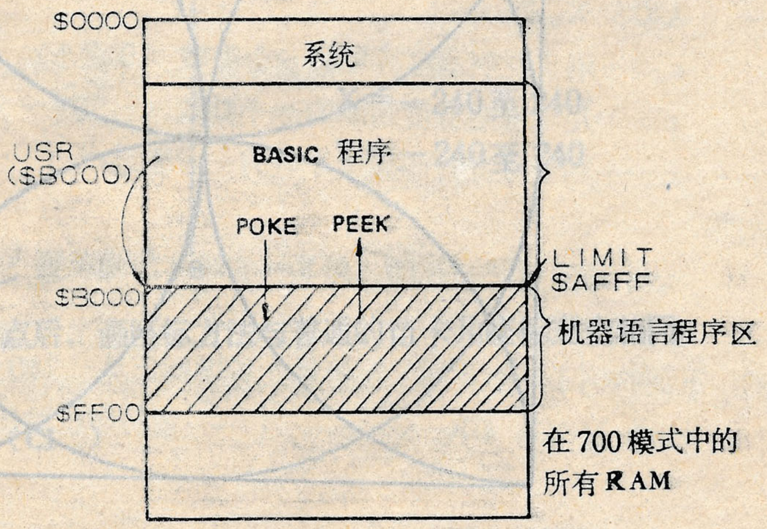
最低级语言（机器语言）能直接地控制CPU而且执行速度快，但用来编制大程序，则要相当熟练的技术。

有了机器语言程序的控制语句，能完成复杂的编程，因为它同时利用了BASIC 语言和机器语言的优点。

利用监控程序或者汇编程序和载入程序，可以产生机器语言程序并且把它载入机器语言程序区(由BASIC的LIMIT语句指定内存区）这样机器语言程序就可以由使用了USR（ ）函数的BASIC程序所调用。用一个以机器代码写出每一步的POKE语句的BASIC程序就可以把机器语言程序调入内存。然后，通过BASIC程序中的USR（ ）函数调用机器语言程序。

在下面的内存图中，右边的略图，表明用POKE和PEEK存取数据和用USR（ ）函数调用机器语言程序。


BASIC
汇编语言
机器语言
CPU
其它高级语言


$0000
系统
USR
($B000)
BASIC 程序
POKE
PEEK
LIMIT
$AFFF
$B000
机器语言程序区
$FF00
在700模式中的
所有RAM

2·8、1　LIMIT……（缩写格式：LIM．）

格式：LIMIT　ad

ad……地址，是一个0到65279的十进制数或者是一个从＄0000到$FEFF的4位十六进制数。

功能：

这个语句用于指定BASIC解释程序可使用的内存区。由ad指定的是BASIC 区的上

限，而这个区之后，即地址（ad＋1）到$FEFF（65279）是机器语言程序或者专用数据区。

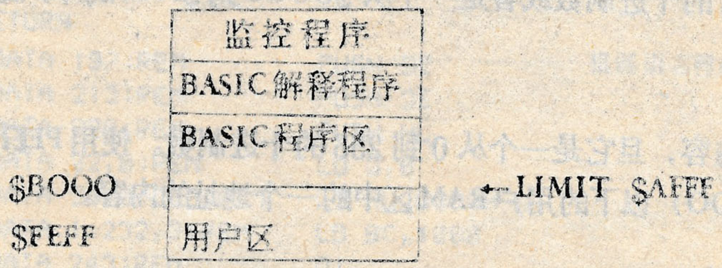
例：

LIMIT　$AFFF

规定BASIC程序区上限地址为$AFFF

注意：

从$FFOO到$FFFF是监控程序的工作区，因此它是不能作为用户区使用的，LIMIT语句必须用在一个BASIC程序的开始位置。



用LIMIT　MAX可以取消由LIMIT　ad设置的地址界限。

2·8、2　POKE

格式：

POKE　ad，d

POKE　@ ad，d

ad……地址，是一个0～65535的十进制数或者是一个从$OOOO到$FFFF的十六进制数。

d……被写的数据：是一个（0到255）的十进制数或者是一个（$OO到$FF）十六进制数。

功能：

这个语句是把数据字节d 写到地址ad中去。

POKE语句能把数据写入内存中的任何存储单元，而完全不顾由LIMIT语句所设置的界限。因此，不小心使用这个语句，可能会破坏监控程序或者BASIC解释程序。

使用POKE@格式是把数据写入53248（$DOOO）以下的用户RAM区中的某一个地址内。

例：　POKE　$DOOO，$5F

POKE　53248，95

上面两个语句执行的效果相同。

注意：

POKE语句指定一个$DOOO之后的地址，并把数据写入这个视屏的RAM区。

2·8、3 PEEK

格式：

PEEK （ad）
PEEK @（ad）

ad……地址是一个0到65535的十进制数或者是一个从$OOOO到$FFFF的十六进制数。

功能：

这个函数得到指定地址的内容，旦它是一个从0到255的十进制数。使用PEEK @格式是得到53248 （$DOOO）以下的用户RAM区中的一个地址的内容。

例如：

下面的程序是显示存在地址40960 （$AOOO）到40975 （$AOOF）中的数据。

```
10 FOR AD＝40960 TO 40975
20 ? PEEK （AD）
30 NEXT AD
```

2·8、4 USR……（缩写格式：U.）

格式：

USR （ad）
USR （ad，X$）
ad……地址（十进制数或者4位十六进制数）
X$……字串数据

功能：

这是一个把控制转移到指定地址开始的机器语言程序中去的特殊函数。正如用CALL ad的情况一样，倘若机器语言程序中包含一个返回指令（RET或RET_CC）的话，控制也返回到USR函数后的语句。

当X$被说明时，在机器语言程序被调用之前，容纳X$的内存区的开始地址就被存入DE寄存器中，而X$的长度则存入B寄存器中，这就使BASIC程序有可能通过串数据转到一个机器语言程序中去。

2·8、5 机器语言程序的编制

在这一节中作为例子出现的是一个用MZ—700所支持的字符充满整个显示屏的机器语言程序。

下面的BASIC程序是把这样一个机器语言程序载入内存并调用它。

```
10 LIMIT $BFFF ································· 规定BASIC区到$BFFF
20 GOSUB 50
30 USR($C000) ·································· 调用机器语言程序
40 END
50 FOR I =49152 TO 49181      ·····:
60 READ M                          :     从DATA语句中读机器语言程序的数据，并把
70 POKE I,M                        :·····
80 NEXT I                     ·····:     它写入机器语言区。
90 RETURN
100 DATA 197:REM          PUSH BC    ··········· 机器语言程序的开始数据
110 DATA 213:REM          PUSH DE
120 DATA 229:REM          PUSH HL
130 DATA 22,0:REM         LD D,0
140 DATA 33,0,208:REM     LD HL,D000H
150 DATA 1,232,3:REM      LD BC,1000
160 DATA 243:REM          DI
170 DATA 211,227:REM      OUT (E3H),A ······ 把储存块转换为视屏RAM（见144页）
180 DATA 114:REM      STO:LD (HL),D ····:
190 DATA 35:REM           INC HL        :
                                        :······ 把一个显示代码置入视屏RAM
200 DATA 20:REM           INC D
210 DATA 11:REM           DEC BC
220 DATA 120:REM          LD A,B
230 DATA 177:REM          OR C
240 DATA 194,14,192:REM   JP NZ,STO
250 DATA 211,225:REM      OUT (E1H),A ······ 把储存块转换为RAM（见168页）
260 DATA 251:REM          EI
270 DATA 225:REM          POP HL
280 DATA 209:REM          POP DE
290 DATA 193:REM          POP BC
300 DATA 201:REM          RET ··············· 返回BASIC程序
```

如果用监控程序产生的机器语言程序并且以文件名DISPLAYCODE存在磁带中的话，那么可利用下面的程序调用机器语言程序。

```
110  LIMIT  $BFFF
110  LOAD  "DISPLAYCODE"
120  USR  ($COOO)
```

第九节　输入／输出（I／O）语句

所有的外部设备（包括软盘驱动器）通过一个任意的普通接口板，就可以与MZ—700系列机相连，为了使用户把MZ—700机与外部设备如X—Y描图仪，纸带穿孔机，音乐组合设备相连可以使用任意的普通接口板。

在普通接口卡上提供的可选地址开关，允许你把0～239（OOH→EFH）的任何接口地址分配给你任何一种外部设备，为了使你临时增加周边设备已留出240—255号地址，用户可以使用INP 和OUT 语句通过任选接口的I／O卡，实现主机与外设之间的数据传送（输入／输出）这些语句的型式如下：

INP ＃P，D……从P口读出8位数，然后把它转换成十进制数，再赋给变量D。

OUT ＃P，D……把变量D的十进制数转换成二进制型式，然后输出到P口。

这些语句极大地扩展了MZ—700系列计算机的功能。

第十节 其它语句

2·10、1 错误处理（ON ERROR GOTO）语句……（缩写格式：ONERR.G.）

格式：

ON ERROR GOTO Lr

Lr……指定行号（错误处理子程序的入口点）

功能：

如果出现错误的话，这个语句将使程序转到行号为Lr的程序行去执行。而IFERN和IFERL语句则是分别根据出现错误类型和出现错误行号，使程序转到开始行号为Lr的捕获子程序去执行子程序，在错误处理子程序中一定要设置一个RESUME语句，以便子程序执行完后，返回到出现错误的行号上去继续执行程序，执行一个ON ERROR GOTO语句时，将会删去由先前的ON ERROR GOTO语句定义的错误捕获行号，用CLR语句亦可删去定义的错误捕获行号。

2·10、2 捕获错误类型（IF ERN）语句

格式：

IF ERN的关系表达式THEN Lr

IF ERN的关系表达式THEN语句

IF ERN的关系表达式GOTO Lr

Lr……指定行号

功能：

当ERN的关系表达式的结果是真时，使程序转移到开始行号为Lr的错误处理子程序去执行子程序或者是执行THEN后面的语句。

ERN是一个专用函数，它能得到对应的错误类型的编号。

例如：

下面所示，如果出现错误程序就转移到1000行开始的子程序去执行子程序，在执行1000行时如果这个错误编号是5的话，又转到1200行开始的子程序去执行子程序。

10 ON ERROR GOTO 1000……说明错误处理子程序行号。

1000 IF ERN＝5 THEN 1200 如果出现的错误是字串溢出错误，则转到1200去。

10 ON ERROR GOTO 50

```
20 INPUT  "X" =; X
30 PRINT SOR〔X〕; TAB〔20〕;  "YES!"; GOTO 20
50 IFERN=3  THEN  70
60 RESUME  20
70 PRINT SOR〔-X〕;  "*I"
80 RESUME  60
90 END
   RUN
   X=9 ▲
    3  YES!
   X=-8 ▲
   2.8284271  *I
```

2·10、3　捕获出现错误行号（IF ERL）语句

格式：

IF ERL的关系表达式THEN Lr
IF ERL的关系表达式THEN语句
IF ERL的关系表达式GOTO Lr
Lr……指定行号

功能：

当ERL的关系表达式结果为真时，程序将转移到开始行号为Lr的子程序去执行子程序或执行THEN后面的语句。

ERL是一个专用函数，用它获得出现错误的行号。

例如：

如果错误出现在250行，下面的语句 使程序转到1300去执行。

```
1010 IF ERL=250 THEN  1300
```

如果错误编号是43并且出现错误的行号不是450时，下面语句使程序返回主程序的520行去执行。

```
1020 IF (ERN=43)*(ERLC> 450) THEN  520
```

2·10、4　返主程序（RESUME）语句……（缩写格式：RESU.）

格式：

RESUME (NEXT)
RESUME Lr

Lr……行号或0

功能：

这个语句是从错误处理子程序返回主程序。

讨论：

系统记住在内存中出现错误的行的编号，并使程序返到该行号去执行或者返回到在纠正错误后另行规定的行号去执行。

RESUME语句有下列四种型式，可随需要选用。

RESUME……返回到出现错误的程序行。

RESUME NEXT……返回到出现错误行的下一行。

RESUME Lr·······返回到Lr行。

RESUME 0……返回到主程序的开始行。

如果没有错误出现遇到RESUME语句时，会产生21类错误(返回错误）。

如果错误处理子程序中没有RESUME语句，则会出现20类错误（CAN'T RESUME ERROR）不能返回错误。

2·10、5 SIZE

格式：

PRINT SIZE

功能：

这是一个特殊函数，得到当前内存中可用于贮存BASIC程序的字节数。

例如：

PRINT SIZE将显示当前内存区的剩余空间（字节数）

2·10、6 PLOT ON……（缩写格式：PL. ON）

格式：PLOT ON

功能：

这个语句把彩绘图打印机当做“显示器用”也即使用了这个语句从键盘上的输入将由打印机印出，因此，MZ—700系列机可以不使用外部显示屏。

这个语句只有安装了彩色绘图打印机和予先执行了MODETN语句后才有效。

例：PLOT ON

注意：

任何没有存在彩色绘图打印机的字符发生器中的字符均以打印一个园点“·”来代之，执行这个语句后 INST ， DEL 键就不起作用了， CTRL + G 键可以用来改变色笔，如果在这种方式中键入程序时，输入结束后，应把 SHIFT 和 BREAK 同时按下，计算机将回到等待命令状态。

2·10、7　PLOT　OFF……（缩写格式：PL.OFF）

格式：

PLOTOFF

功能：

这个语句将取消绘图打印机的PLOTON方式。

例如：

PLOTOFF

2·10、8　CONSOLE……（缩写格式：CONS.）

格式：

CONSOLE （IS,Ln <，Cs，Cn >）

Is：卷起区的首行

In：卷起区的行数

Cs：卷起区的起始列

Cn：卷起区的列数

例如：

```
CONSOLE O，25，0，40
CONSOLE 5，15
CONSOLE 0，25，5，30
CONSOLE 0，10，0，10
CONSOLE
```

功能：

这个语句用来规定卷起区大小，也即这个区可以用PRINT "$\boxed{C}$"来清。

上面第一个例是规定全屏做为卷起区。第二个例规定5～20行为卷起区。第三个例规定5～35列为卷起区。第四个例规定屏幕左上角10＊10的位置为卷起区。为了隔开显示区左边或右边的图像，这个语句是很有用的，这时图像被隐藏在屏幕的这一边。

上面最后一个例未规定卷起区，当不规定卷起区时，可以上卷，可以下卷。然而，这会使屏幕编辑产生困难，因为Cn和In的值实在太少了。

第十一节　监控程序的作用

在BASIC解释程序的输入／输出控制系统（IOCS）部分含有一个监控程序，从而使它输入机器语言程序十分容易。这个监控程序用内存地址FFOOH到FFFFH的区域作为堆栈区。

该监控程序含有类似于BASIC的屏幕编辑功能，这就使它能够改变像下面描述的64 K RAM区内的任何地址的内容。

2·11、1　编辑格式

：address＝data　data　data

：（冒号）……指明该行以下可以被编辑。

地址　……指明内容可以改变的内存区的起始地址。（四位十六进制数字）

＝　……地址与数据的分隔符。

data……一个2位十六进制数或者一个分号（；）加到写入说明的地址的字符中去，空格是用来分隔数据项的。

2·11、2　打印机开关命令……（P命令）

格式：

＊P

这个命令与D或F命令一块使用，可以从打印机或者显示屏上输出数据。如果打印机未与计算机连接，则会显示“ERR？”信息，然后，监控程序处于等待输入别的命令状态。此时应检查打印机的连接或者再次执行P命令把输出设备换为显示屏。

2·11、3　转储命令　……（D命令）

格式：

＊D＜起始地址＜␣末地址＞＞　.

这个命令是将存贮器中首地址内容转移到末地址中去。如果末地址被省略，则从说明的地址开始转储128个字节储存块的内容。如果首末地址均被省略．它就予先转储128个字节块以下内容。转储这种数据格式如下：

$$:\ \underset{\substack{\uparrow\\ \text{始地址}}}{HHHH}=\underbrace{HH\ HH\ HH\ HH\ \ HH\ HH\ HH\ HH}_{\text{8个字节（16进制代码）}}\ /\ \underbrace{ABCDE.G.}_{\text{8个字节（字符）}}$$

移动光标到相应的字节上，可以改变任何地址的内容，操作是键入新的数据，然后按CR键。

注意：

在字符数据域中，控制代码用园点（·）显示。按BREAK键就停止转储输出，而按SHIFT和BREAK键则立即使监控程序返回输入命令状态。

2·11、4　内存设置命令

格式：

＊M〔始地址〕

这个命令是改变存储器的内容，如果首地址被省略，该地址指的就是当前程序计数器所设定的地址。把SHIFT和BREAK键一块按下，则会终止这个命令。

当键入这个命令后，存储块的首地址和它的内容，便按前述的编辑格式显示出来，然后把光标移到待改的数据上。键入新的数据和按下 CR 键，接着显示出后面的地址和它的内容。

2·11、5　检索命令……（F命令）

格式：

＊F〔首地址〕⎵〔末地址〕⎵〔数据〕⎵〔数据〕⎵……

这个命令对存储区数据串从首地址到末地址进行检索。一旦找到，该串的地址和它的内容就显示到屏幕上。同时按下SHIFT和BREAK便终止这个命令。

2·11、6　调用子程序……（G命令）

格式：

＊G〔调用地址〕

这个命令是调用指定地址开始的子程序。栈指针指向FFEEH地址。

2·11、7　转移命令……（T命令）

格式：

＊T〔始地址〕⎵〔末地址〕⎵〔目的地址〕

这个命令是把首地址到末地址间的内容传送到目的地址的存储区。

2·11、8　存带命令……（S命令）

格式：

＊S〔始地址〕⎵〔末地址〕⎵〔执行地址〕：〔文件名〕

这个命令是把首地址到末地址之间的存储区的内容存入磁带。内容置于文件名之后。

2·11、9　载入命令 ……（L命令）

格式：

＊L ＜载入地址＞ ＜：文件名＞

这个命令是把指定的文件从载入地址开始载入存储区。载入地址省略，则该文件中包含的执行地址就设定为载入地址，若文件名省去，则把在磁带中首先遇到的文件载入内存。要是检查出错误或者在执行期间按下BAERK键，将会显示“ERR？”信息。然后监控程序返回到等待输入命令状态。否则命令执行通过。此时计算机便回到等待输入命令状态。

2·11、10　校对命令 ……（V命令）

格式：

＊V〔文件名〕

这个命令是从磁带中读出指定的文件，然后和内存中的内容进行比较，以保证用存带命令录制的程序的正确性。若从磁带中读出的数据与内存中的内容不符，便显示“Err？”信息。

2·11、11　返回命令 ……（R命令）

格式：

＊R

该命令使控制返回到调用监控程序时的系统程序，并且把SP（栈指针）和HL寄存器的内容恢复到调用监控程序初值。通过执行BYE命令亦可重新恢复。

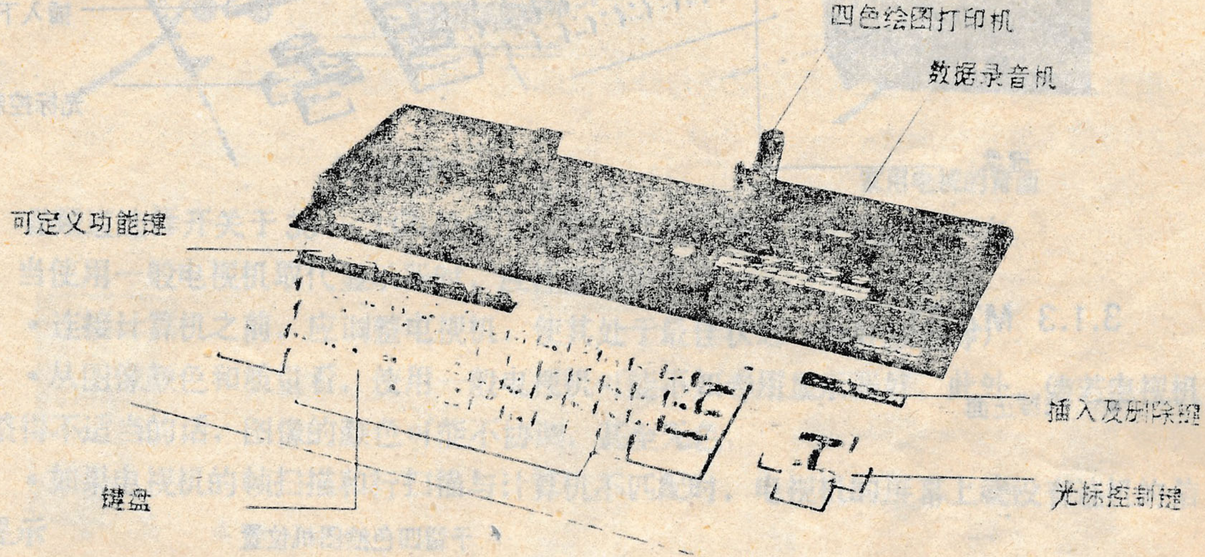
如果监控程序调用指针在FFOOH到FFFFH之间的系统程序或者栈指针并不包含返回地址时，这个命令就不能使控制返回。在这种情况下，用G命令可以访问热启动的入口点。

第三章　MZ— 700的操作

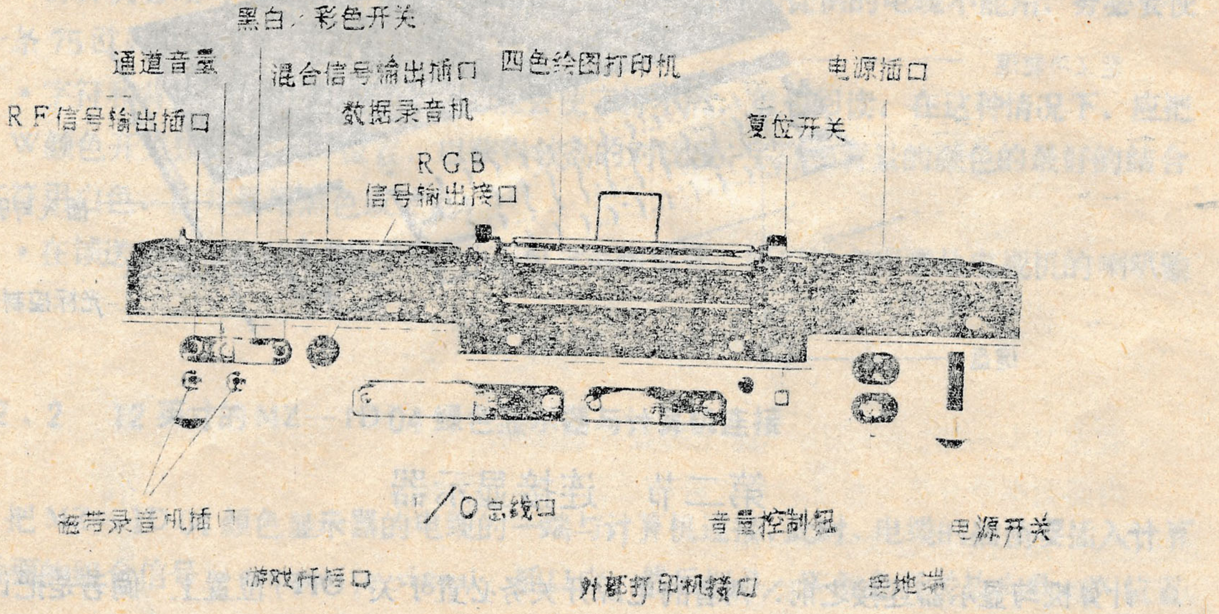
第一节　MZ— 700系列个人计算机的外貌

3·1．1　MZ—731

- 机器正面

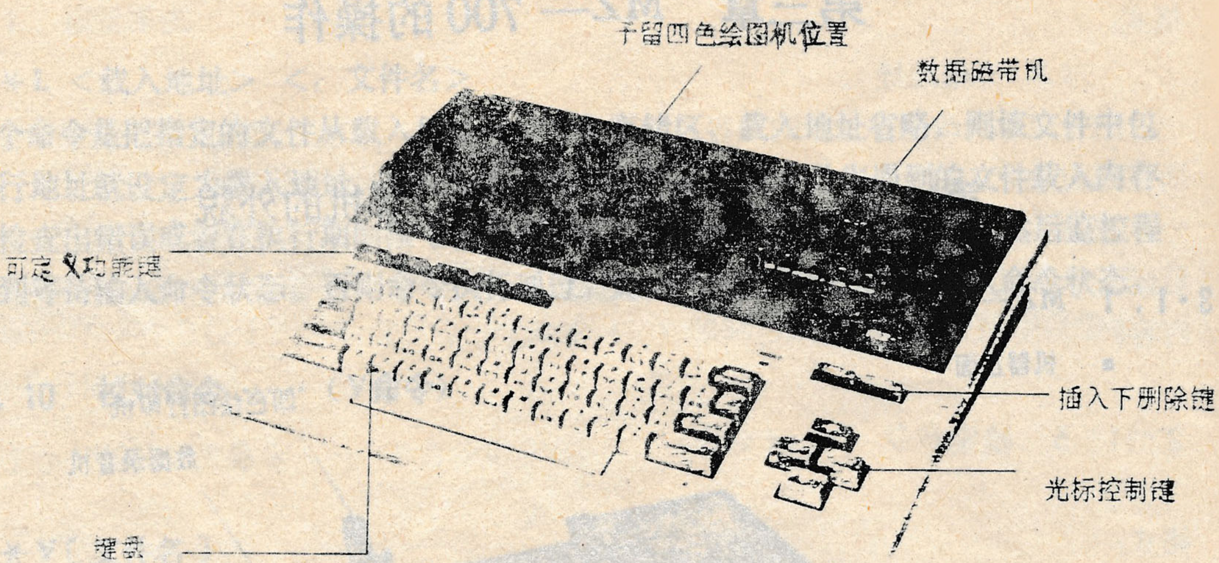

四色绘图打印机
数据录音机
可定义功能键
插入及删除键
光标控制键
键盘

- 机器后背面


黑白/彩色开关
通道音量
混合信号输出插口
四色绘图打印机
电源插口
RF信号输出插口
数据录音机
复位开关
RGB
信号输出接口
磁带录音机插口
I/O总线口
音量控制钮
电源开关
游戏杆插口
外部打印机接口
接地端

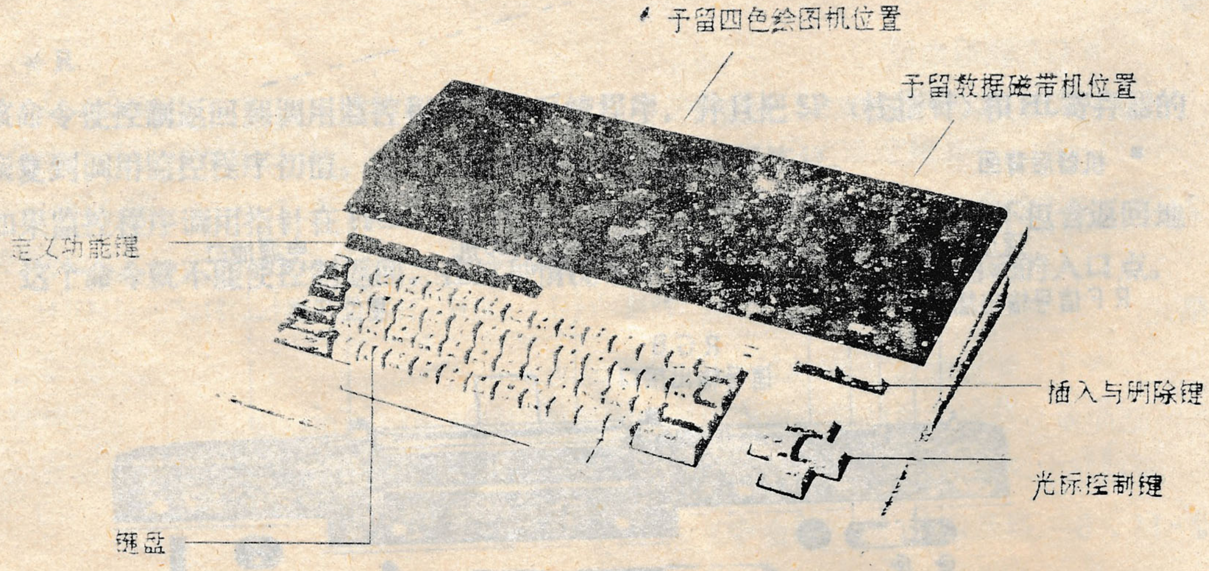
3.1.2 MZ-721

■ 机器正面


予留四色绘图机位置
数据磁带机
可定义功能键
插入下删除键
光标控制键
键盘

3.1.3 MZ-711

■ 机器正面


予留四色绘图机位置
予留数据磁带机位置
定义功能键
插入与删除键
光标控制键
键盘

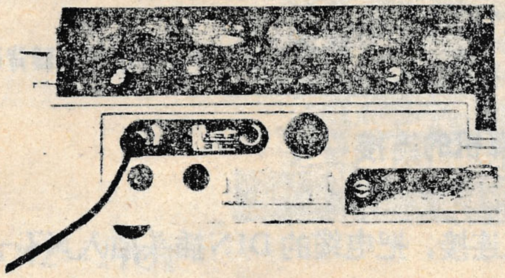
第二节 连接显示器

计算机与显示器连接之前，两者的电源开关务必置于关（OFF）位置上．倘若是把计算

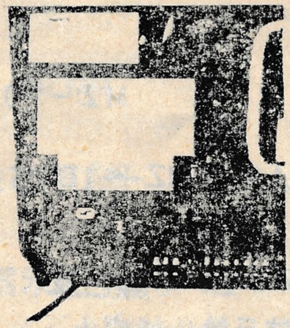
机与电视（TV）连接，则先要断开电视机的天线。

3·2．1　电视机与MZ—700机的连接

从电视机的UHF天线接头上拆开电线。把连接电缆的插销插入计算机后面的RF 信号输出插口，而另一端的插销则插到电视机的75欧姆的UHF天线插座上。



机器后背面



家用电视的背面

置频道选择开关于36 ± 3Ch位置上，这一点无论在哪一个地区都不变。

当使用一般电视机取代显示器时，应注意如下几点：

•连接计算机之前，应调整电视机，使其处于最佳状态（调好颜色等）。

•从图像颜色和质量看，使用一般电视机可能不如专用显示器好。此外，倘若电视机调整得不适当的话，图像的颜色可能不协调，甚至无色。

•如果电视机的帧扫描和行扫描与计算机不匹配时，电视机的屏幕上就没有连机的信息显示。

•与计算机连接之前，务必要拆去电视机的天线，否则，从天线幅射出来的信号，可能会干扰别的电视机。

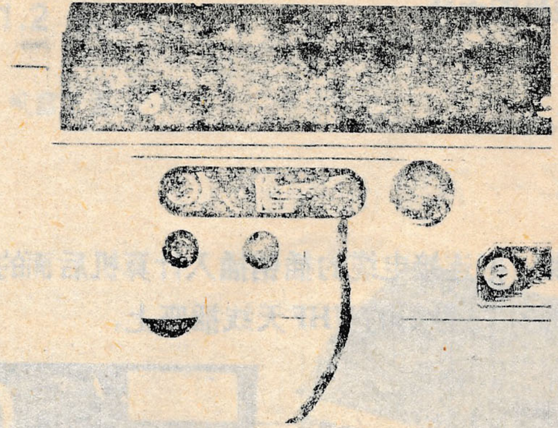
•计算机必须与电视机的75欧姆的天线端点连接，如果提供的电缆不能用，务必要使用一条75欧姆的同轴电缆。

•字符和背景颜色的某些结合，可能会使字符不清，难于阅读，在这种情况下，应把B／W颜色开关拔向B／W位置，以获得较高的对比度，字符和背景的颜色的最好的结合是字符用白色，而背景用黑色或兰色。

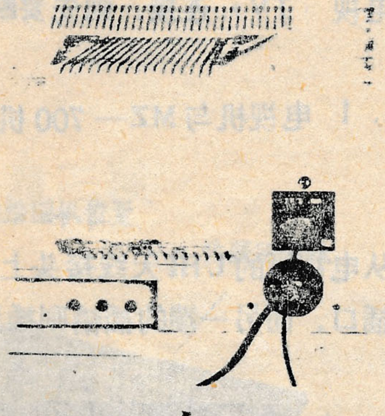
•在馈送到电视机的RF信号中，并不包含声频信号，所以没有声音从电视机的喇叭输出。

3·2、2　12英寸的MZ—1D04绿色显示器与计算机连接

把MZ—1D04颜色显示器的电缆的一端与计算机连接，此时，电缆的插销要插入计算机后面的组合信号（composite signal）插口中，然后把B／W颜色开关拔向B／W位置



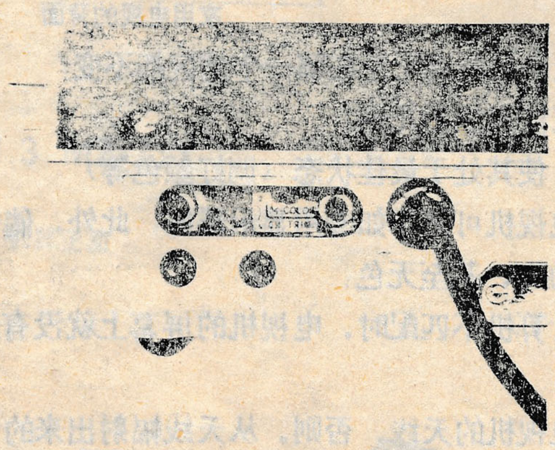
MZ－700系列电脑后背面



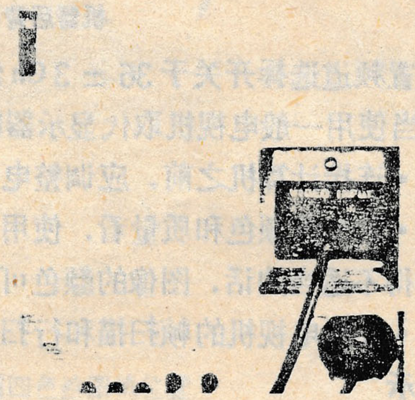
MZ－1D04专用显示器背面

3·2、3 14英寸MZ—1D05彩色显示器与计算机的连接

用MZ—1D05颜色显示器的电缆与计算机连接，把电缆的DIN插头插入MZ—700机的RGB信号输出插座上。

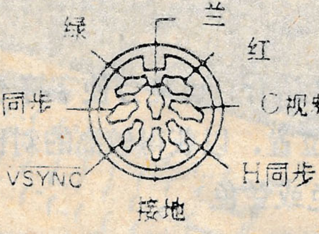


MZ－700系列电脑后背面



MZ－1D05专用显示器背面

RGB信号输出接口中各针的功能如下



DIN插座的RGB信号输出（从背面看）

第三节　数据录音机

- 装在MZ—731和MZ—721上的数据录音机

内部数据录音机与普通的盒式磁带录音机的操作相同.

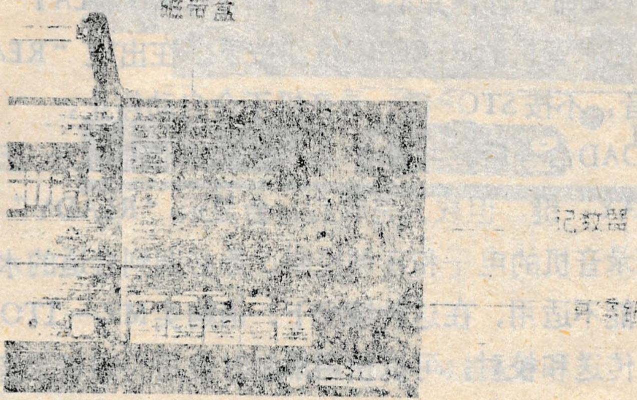
RECORD 按这个键是录制程序和数据.

PLAY 按这个键是载入程序和数据

REWIND　按这个键是反绕磁带

FFWD　按这个键是快速绕带

STOP／EJECT　按这个键是停止磁带，当载入或录制程序或数据后，磁带停止时，按这个键的作用就是释键或弹出磁带盒



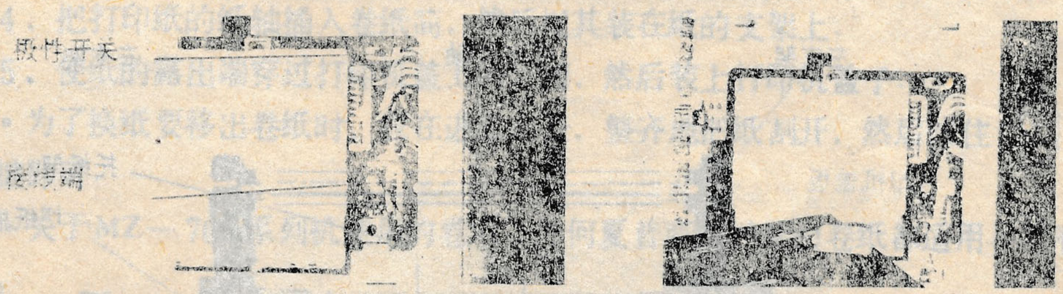
■MZ—1TO1

MZ—1TO1数据录音机，可以装在MZ—711机上，安装过程如下：

（1）关掉计算机的电源开关从AC的引出端拔出电源电缆

（2）取出位于背面板左边的二个螺丝，然后移开数据录音机的盒盖

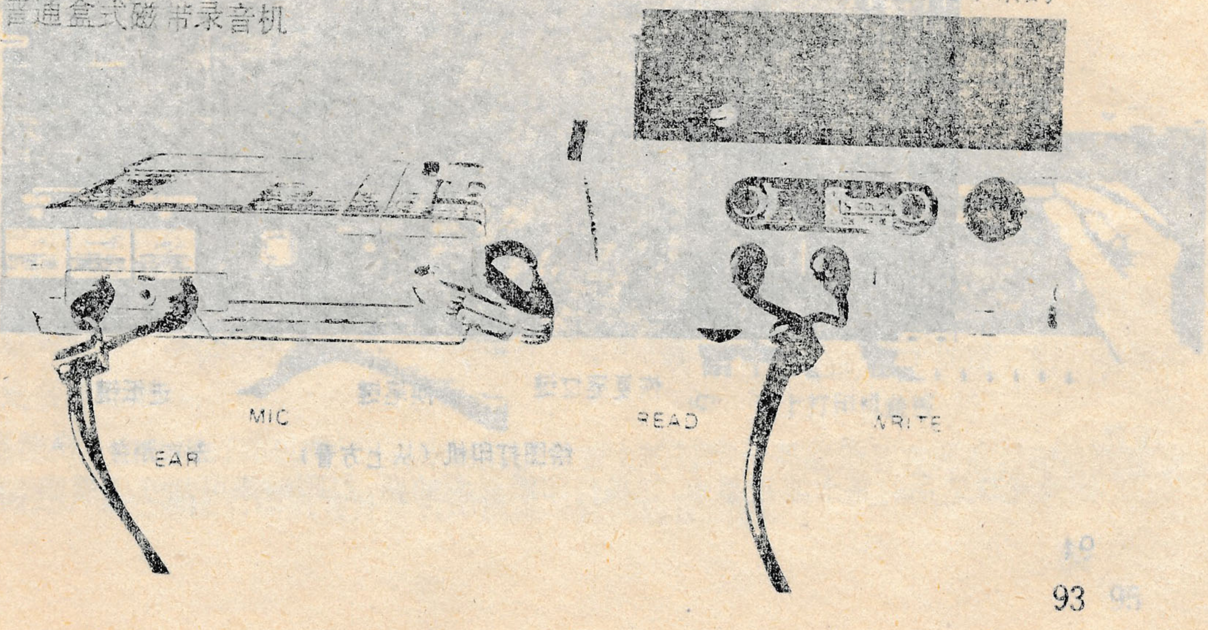
（3）移开接线插座的盖子，



（4）把MZ—1TO1的插销插入MZ—711机的录音机箱后面左边的9针插座内

（5）把该数据录音机放入录音机箱内，然后用二个螺丝钉加固在这样做时，应小心以避免挂住录音机和计算机之间的连接电缆，（否则螺丝钉是上不紧的）

■普通盒式磁带录音机



使用工业上通用的3.5mm微型插销音频电缆，计算机的WRITE（写）插孔和盒式磁带机的MIC插口连接，而把计算机的READ（读）插孔与盒式磁带机的EXT SP或EAR插口连接

使用普通盒式磁带机时应注意如下几点：

（1）当键入SAVE命令时，并不显示“↓RECORD PLAY”等字样，因此，在键入这个命令之前，务必把录音机的RECORD按纽按下，在出现“READY”信息后，应按下STOP键，以停止录音，不按STOP键，录音机不会自动停止。

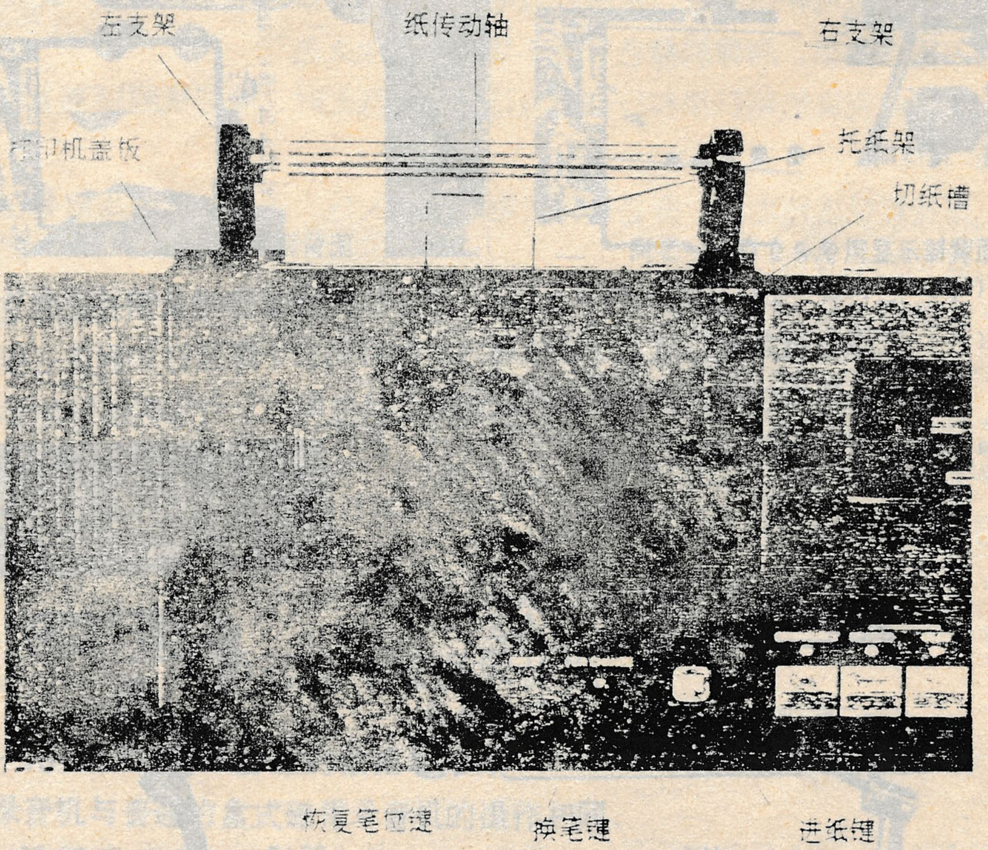
（2）当键入LOAD命令后，并没有“↓PLAY”信息显示，因此使用者键入这个命令后，接着应按下“PLAY”键，但载入完成后，会显示“READY”信息。

（3）盒式磁带录音机的电子和音调控制，务必调到合适的水平上。一些盒式录音机（具有自动调平）可能不适用，在这种情况下，请购买MZ—1T01。

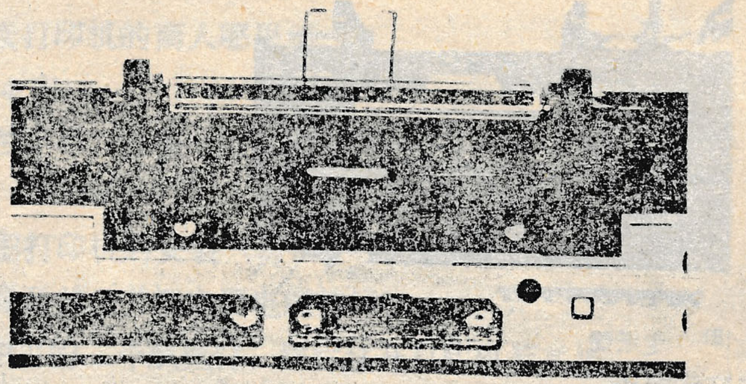
（4）对于任何传送和校对，可以使用录制时使用的磁带录音机，如果传送或校对用的磁带录音机，不是录制时用的录音机，那么可能会出现既不能传送也不能校对的毛病。

（5）插头的极性不对，可能无法将程序调入计算机。此时试着设置极性开关。用推荐您使用MZ－1T01。

第四节　彩色绘图打印机



绘图打印机（从上方看）



纸进口处

（绘图打印机（从后面看）

■装卷形打印纸

1、移出打印机盖子

2、把卷纸的一端切割整齐，然后把它插入打印纸的引纸槽内（这样做时，务必小心，避免纸折叠起皱）。

3、接通 MZ—731 的电源开关， 然后按下 $\boxed{\downarrow}$ 进纸键 纸将向前方推出，直到纸的顶端高出出口 3～5Cm时，才释放 $\boxed{\downarrow}$ 键。

4、把打印纸的纸轴插入卷纸筒，然后把其装在纸的支架上

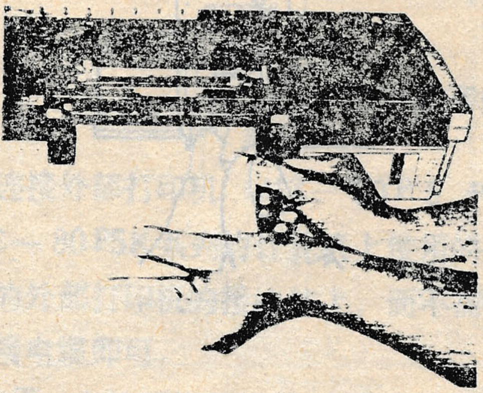
5、使纸的露出端穿过打印机盖上的切槽，然后装上打印机盖子

•为了换纸要移出卷纸时，应在进纸口处，整齐地把纸割开，然后按住 $\boxed{\downarrow}$ （进纸）键即可

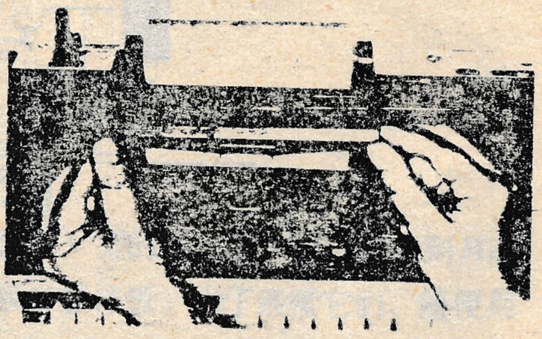
•关于MZ—700 系列机使用的卷纸，任何夏普商人经营的卷纸都适用 但不要用其他纸。

纸长 23～25 米，可装卷纸的最大直径是 50 毫米。如果使用的卷纸直径大于 50 毫米，则会影响纸的正常馈送，结果使打印质量难于保证。

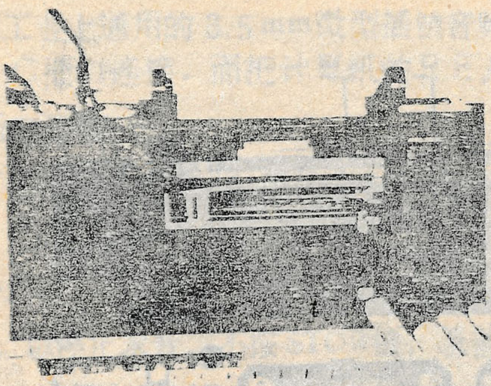
Procedures for loading roll paper 按装滚筒纸方法



(A) 装纸方法



(C) 盖上打印机盖板



(8)　走纸键

■装笔和换纸

1）取出打印机盖，用元珠笔或类似的针状物捅一下换笔开关，这使笔架从打印机的左边向右边移。

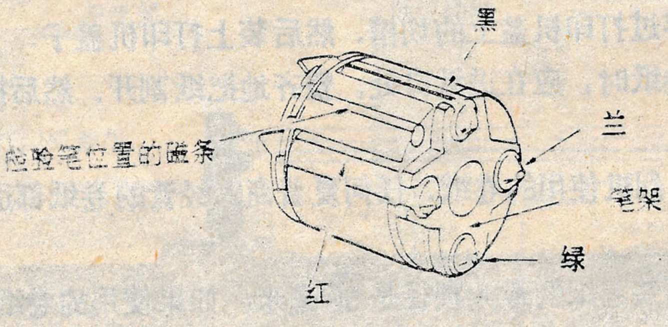
2）按下弹笔杆，笔便从夹架中弹出，不过在做这一步之前，应先把你的手指放在笔的上方，以防止按弹笔杆时，把笔完全弹出掉入打印机内。

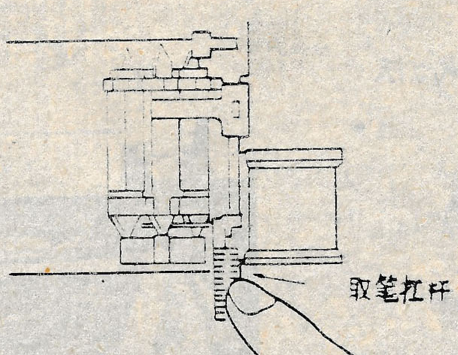
3）插入新笔。

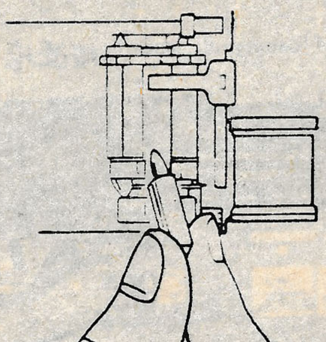
4）再按换笔键，使另一支色笔转到笔架的顶部。

5）更换四支笔（黑色、兰色、绿色、红色）都用同样方法。换完最后一支笔时，应按RESET开关以使打印机处于准备用黑色笔打印的初始状态。

此时可以执行BASIC的TEST命令，以验四支笔是否打印正常。







注意:

因为元珠笔用的是水溶液墨水，如果打印纸受潮的话，打印结果可能变得模糊难辨，因

此应妥善保管打印纸，以免受潮。

■要换笔的话，可到卖打印机的商人哪里买

- EA—850B（黑色4支）
- EA—850C（黑、兰、绿、红各一支）

■MZ—1PO1

MZ—1PO1彩色绘图打印机的安装（与MZ—731的模式不同）。

1）关掉计算机的电源开关，并把电源电缆拔掉。

2）取出位于后面板中央的二个螺丝钉，然后移开打印机箱盖板。

3）务必保证把印刷线路插件板上的打印开关置于INT位置上。

4）把打印机的插头插入印刷线路插件板上相匹配的插座内，然后把打印机装入打印机箱，并用二个螺丝钉紧固好，这样做时，要小心些，以免挡住数据录音机和计算机间的连接电缆（否则螺丝钉就不能上紧）。

打印机接口
打印机开关
电源接头

外部打印机 ←→ INT（绘图打印机）

彩色绘图打印机的连接　　　MZ-700

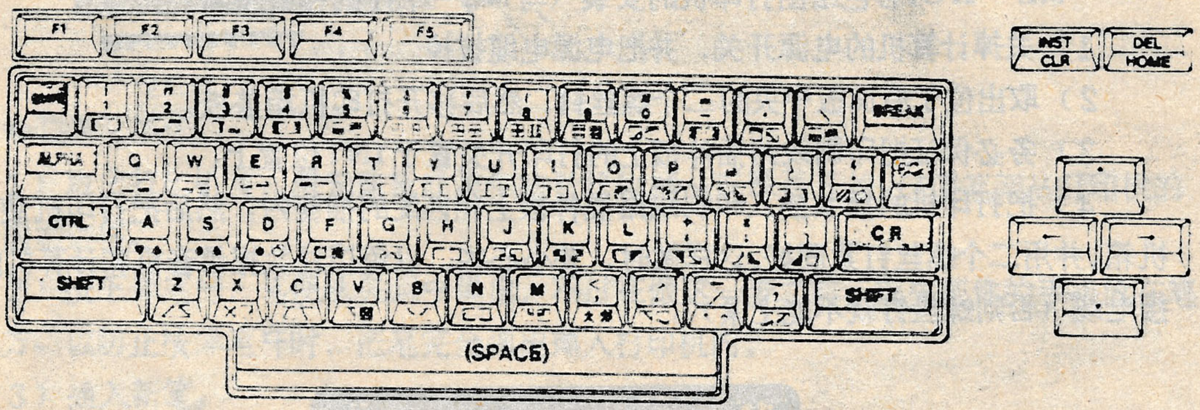
■连接外部打印机（MZ—80P5（K））

MZ—80P5K系列的计算机上使用的MZ—80P5(K)型的打印机，可以连接到MZ—700的外部打印机的接线柱上，而不需任何专用的接口插件板。为了连接它们，随便用一条连接电缆即可。

当使用一台外部打印机时，印刷电路的插件板上的打印开关必须放在外部打印机的位置上。因此，彩色绘图打印机与外部打印机是不能同时使用的。

应注意的是：假如你运行一个含有彩色绘图打印机语言程序，而使用的是外部打印机的话，那么，对应于绘图打印机的控制代码，将以毫无意义的字符印出

第五节　键盘操作


F1
F2
F3
F4
F5
INST
CLR
DEL
HOME
BREAK
ALPHA
Q W E R T Y U I O P
CTRL
A S D F G H J K L
CR
SHIFT
Z X C V B N M
SHIFT
(SPACE)

3·5、1

打字键盘

除特殊控制键外，键盘上的每一个键只代表特定的字符，当一个键被按下时，输入的字符取决于所选择的输入方式。

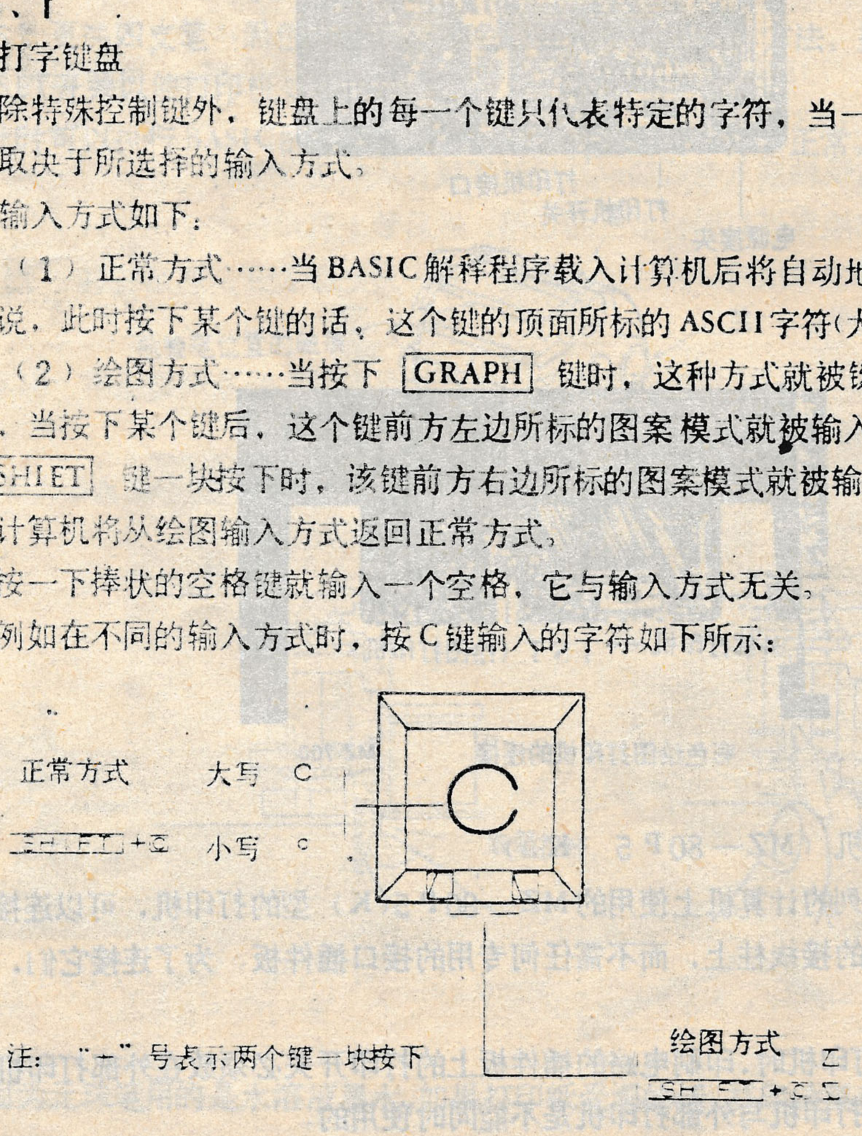
输入方式如下：

（1）正常方式……当BASIC解释程序载入计算机后将自动地置于这种输入方式，也就是说，此时按下某个键的话，这个键的顶面所标的ASCII字符（大写和小写）就被输入。

（2）绘图方式……当按下 GRAPH 键时，这种方式就被键入计算机，使用绘图方式时，当按下某个键后，这个键前方左边所标的图案模式就被输入计算机。　当把这个键和 SHIET 键一块按下时，该键前方右边所标的图案模式就被输入计算机，按下ALPHA键，计算机将从绘图输入方式返回正常方式。

按一下棒状的空格键就输入一个空格，它与输入方式无关。

例如在不同的输入方式时，按C键输入的字符如下所示：


正常方式
大写　C
SHIFT +C
小写　c
C
绘图方式
SHIFT +C

注：“+”号表示两个键一块按下

特殊键说明如下：

SHIFT 按下这个键可以改变输入字符的状态。

对字母键而言，按下 SHIFT 键再按某个字母键，则输入的字是小写字母，对不是字母键的其它键来说，按下这个键再按某个键时，则输入计算机的是这个键顶部最上一排所标的符号，在绘图方式时，按下这个键，再按某个键时，则输入计算机的是这个键前方右边所标的图案模式。

CR 按下这个键将输入 CR （回车）代码，作用是结束一程序行的输入，并把光标移到下一行的开始位置。

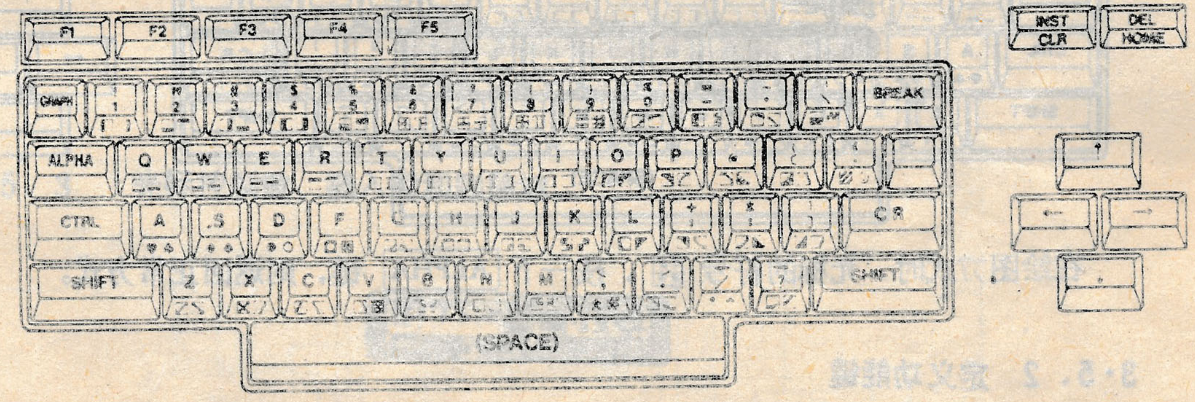
BREAK 按下此键就输入一个BREAK（中断）代码，把它和 SHIFT 一块按下时，将会暂停程序执行或者数据录音机的运转。

GRAPH 按下这个键就把正常方式改为绘图方式，也就是说按这个键后，再按某个键时，输入计算机的是这个键前方左边所标的图案模式。

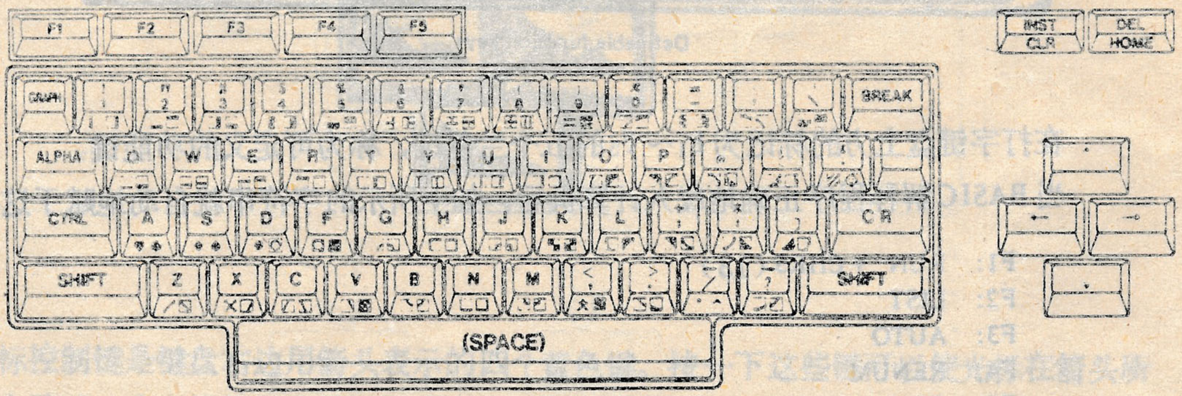
ALPHA 按下这个键把绘图方式改为正常方式，正常方式光标的符号为" # "而绘图方式时光标符为" ▩ "。

（1）正常方式（字母数字方式）

正如下图所示，在正常方式时，通过每个键输入的字符是指阴影所衬托的字符。

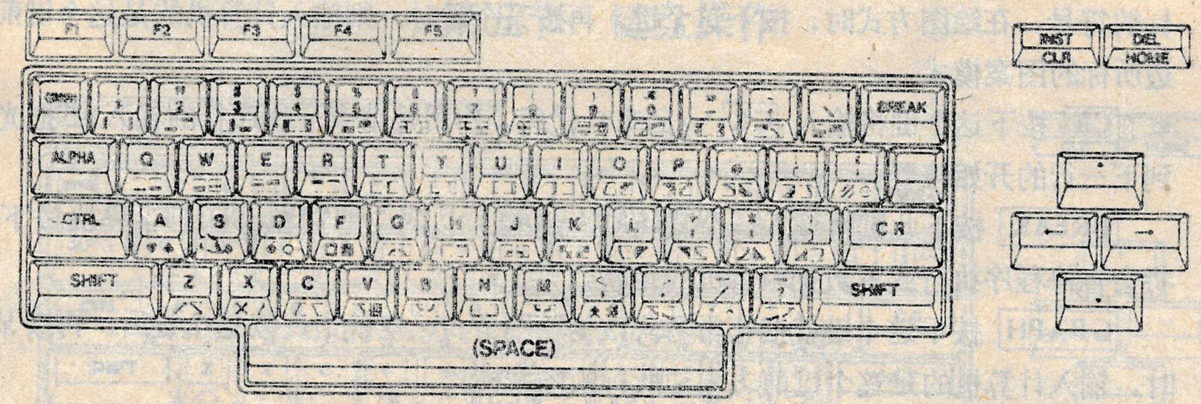

F1
F2
F3
F4
F5
INST CLR
DEL HOME
GRAPH
BREAK
ALPHA
Q W E R T Y U I O P
CTRL
A S D F G H J K L
CR
SHIFT
Z X C V B N M
SHIFT
(SPACE)

当 SHIFT 键与别的键一块按下时，则输入的是小写字母，或图中阴影所衬托的其它符号。

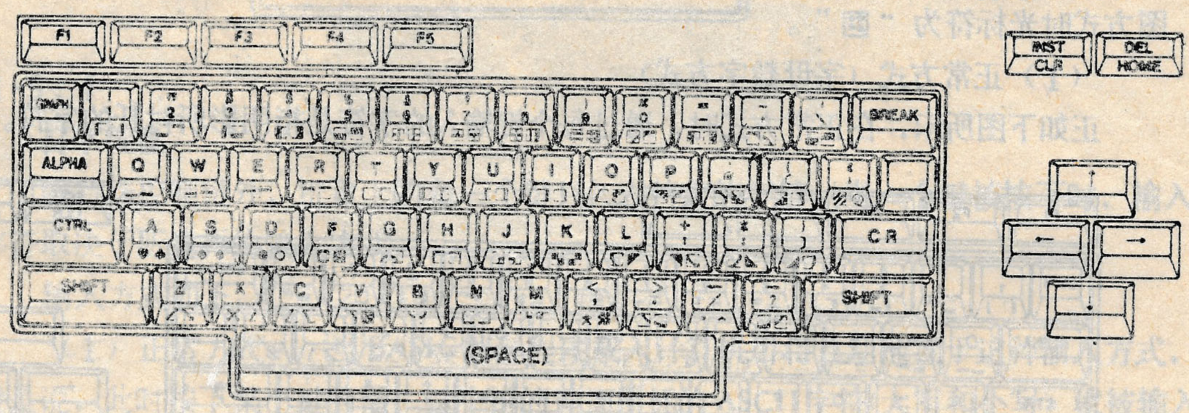

F1
F2
F3
F4
F5
INST CLR
DEL HOME
GRAPH
BREAK
ALPHA
Q W E R T Y U I O P
CTRL
A S D F G H J K L
CR
SHIFT
Z X C V B N M
SHIFT
(SPACE)

（2）绘图方式：

按下 GRAPH 键，就自动地把计算机置于绘图输入方式。在绘图方式时，通过每个键所输入的绘图模式指的是图中阴影所衬托的图案。在这种方式中，按任一光标控制键INST／CLR或DEL／HOME键，将分别地输入 ↑ ← → ↓ C 或 H 字符。

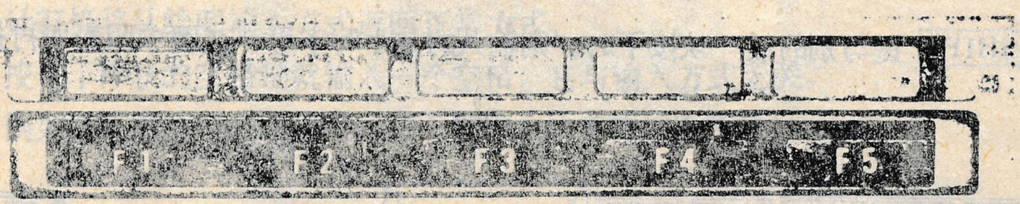

F1
F2
F3
F4
F5
INST CLR
DEL HOME
GRAPH
BREAK
ALPHA
CTRL
CR
SHIFT
SHIFT
(SPACE)

当 SHIFT 键与别的键一块按下时，则输入的是图中阴影所衬托的符号。


F1
F2
F3
F4
F5
INST CLR
DEL HOME
GRAPH
BREAK
ALPHA
CTRL
CR
SHIFT
SHIFT
(SPACE)

在绘图方式时，光标的符号 ▦ ，按一下 ALPHA 键，则返回正常方式。

3·5、2 定义功能键


F1
F2
F3
F4
F5

Definable function keys

在打字键盘上方的标记为F_1～F_5的五个兰色键，称为可定义的功能键。

当BASIC解释程序正确地载入计算机后，如下所示的字符串就自动地赋于这些键。

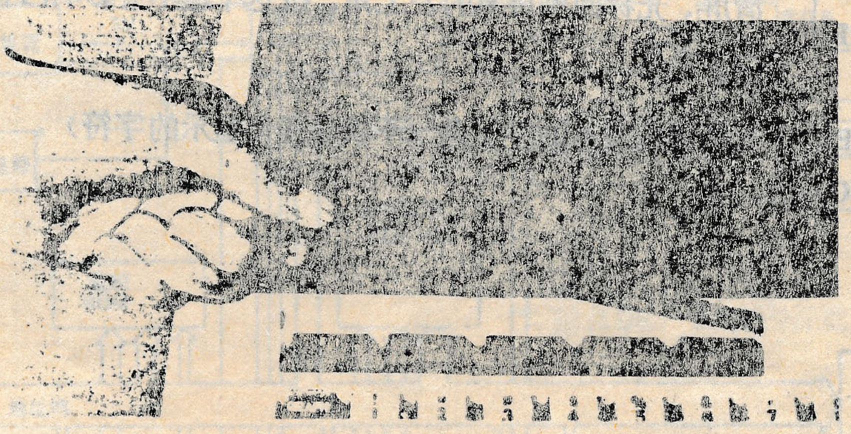
```
F1:  RUN  +CHR$(13)
F2:  LIST
F3:  AUTO
F4:  RENUM
F5:  COLOR
```

SHIFT + F1: "CHR$ ("
SHIFT + F2: "DEF KEY ("
SHIFT + F3: "CONT"
SHIFT + F4: "SAVE"
SHIFT + F5: "LOAD"

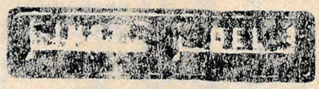
当这些键之一被按下时，赋于这个键的字符串将被输入计算机。因此在输入时，如果经常遇到赋于这些键的字串的话，只要按相应的功能键即可。通过DEFKEY语句，可以把你认为常用的字符串赋给任一个可定义的功能键。

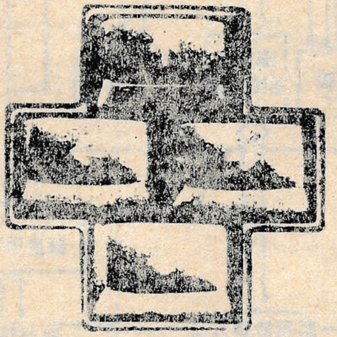
▬可定义功能键的标记

把赋于这些功能键的字串写成标签放在这些键上方的透明盖子之下。这样操作时将十分方便。如下图所示透明盖子是很容易移开的。



2·5、3　光标控制键、插入键和删除键





光标控制键是键盘右边用箭头表示的四个黄色键。按一下这些键可以使光标在箭头所指的方向移动一个字符位置，编辑程序时，可用这些键。

INST CLR 和 DEL HOME 键功能如下：

INST CLR 在光标所在位置插入一个空格，此时后面所有的字符均向右移动一个位置。INST：插入。

DEL HOME 擦掉光标左边的字符，并且所有后面的字符均向左移动一个位置。

DEL：即删除。

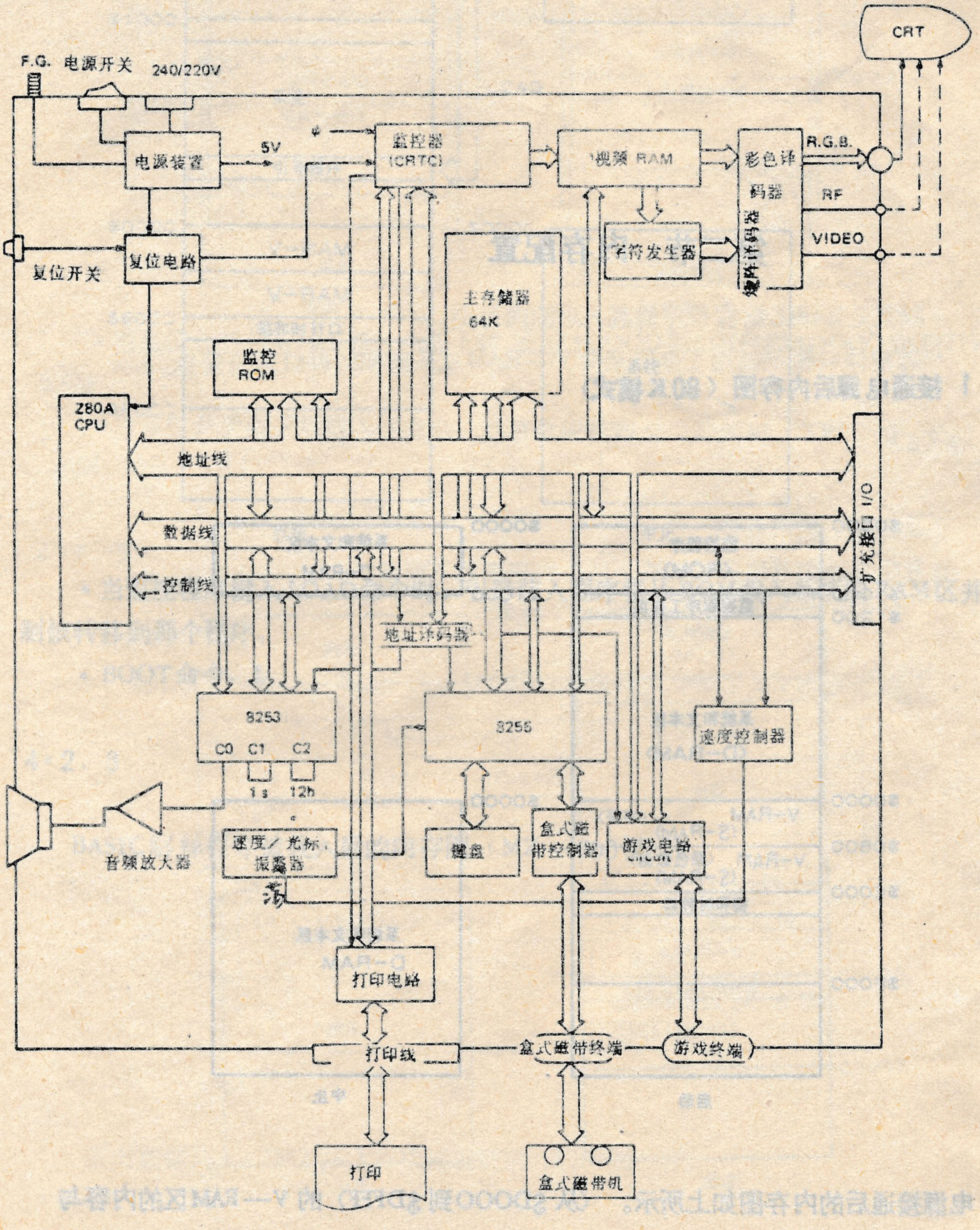
SHIFT + INST CLR 清屏，光标回到屏幕的左上角，按这个键不会影响内存中的程序。CLR是清。

SHIFT + DEL HOME 光标回到屏幕的左上角（不映响任何显示的字符）。

第四章　硬件

第一节　MZ—700系统图

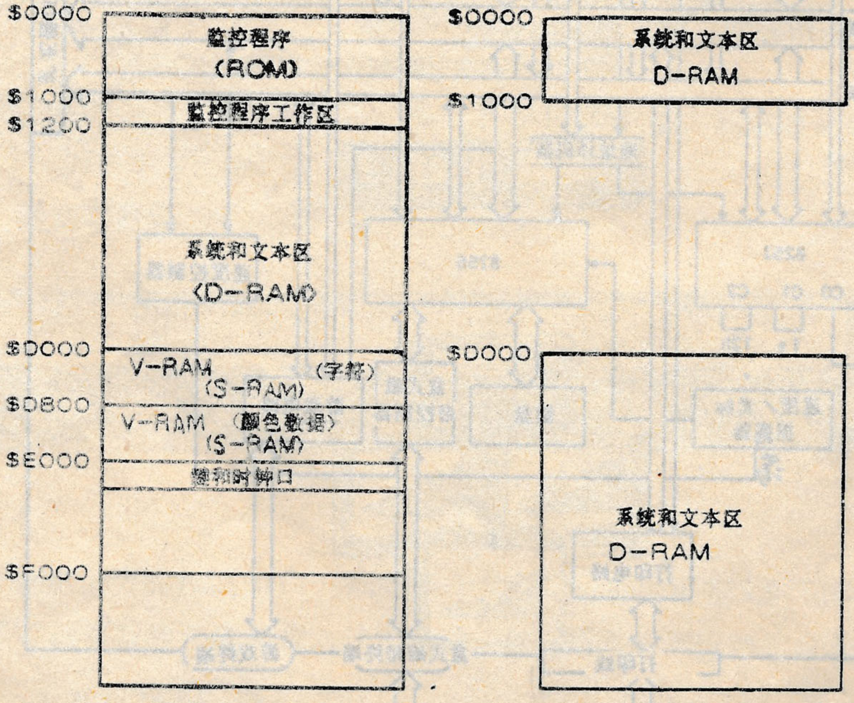
MZ—700系列计算机的系统接线框图



CRT
F.G.
电源开关
240/220V
电源装置
5V
φ
监控器 (CRTC)
视频 RAM
彩色译码器
R.G.B.
RF
VIDEO
字符发生器
矩阵译码器
复位开关
复位电路
主存储器 64K
监控 ROM
Z80A CPU
地址线
数据线
控制线
扩充接口 I/O
地址译码器
8253
C0
C1
C2
1s
12h
8255
速度控制器
音频放大器
速度/光标振荡器
键盘
盒式磁带控制器
游戏电路
打印电路
打印线
盒式磁带终端
游戏终端
打印
盒式磁带机

第二节　内存配置

4·2、1 接通电源后内存图（80K模式）


$0000
监控程序
(ROM)
$1000
监控程序工作区
$1200
系统和文本区
(D-RAM)
$D000
V-RAM
(字符)
(S-RAM)
$D800
V-RAM
(颜色数据)
(S-RAM)
$E000
键和时钟口
$F000
$0000
系统和文本区
D-RAM
$1000
$D000
系统和文本区
D-RAM

启动　　　　中止

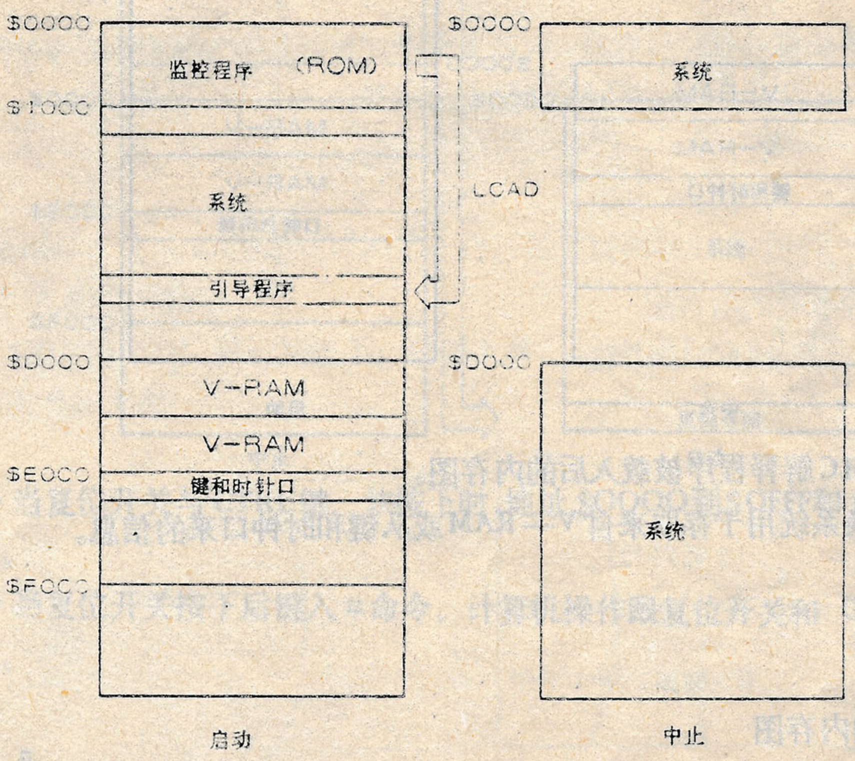
• 电源接通后的内存图如上所示。（从 $DOOO到 $DFFF）的 V—RAM区的内容与

MZ—80K不同）。

- 监控程序ROM的入口点与MZ—80K同用。

4·2、2

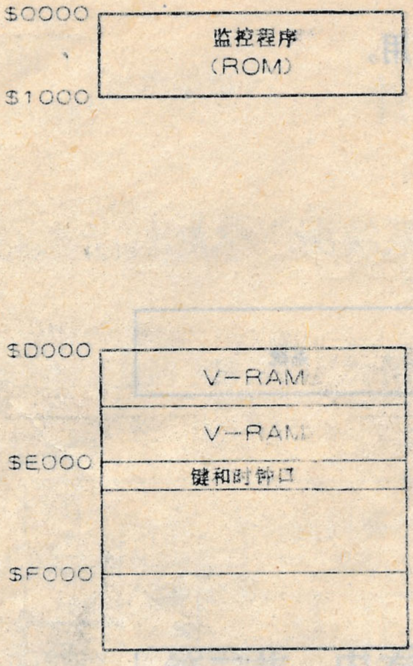
载入系统程序（BASIC）时的内存图


$0000
监控程序 (ROM)
$1000
系统
引导程序
$D000
V-RAM
V-RAM
$E000
键和时钟口
$F000
LOAD
$0000
系统
$D000
系统
启动
中止

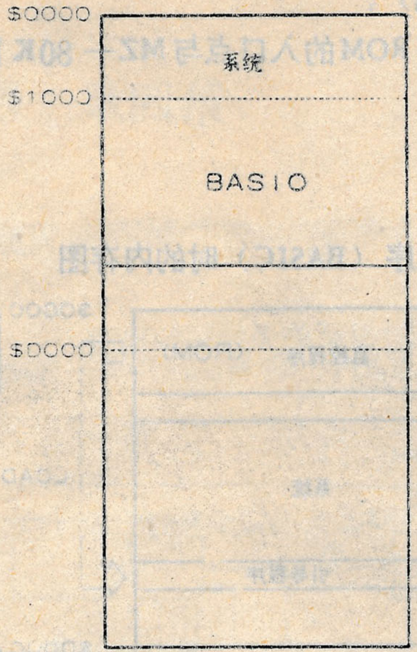
- 当监控程序键入LOAD命令时，引导装入程序将从ROM载入系统的RAM区并且控制被转移到那个程序。
- BOOT命令：L

4·2、3

BASIC解释程序被载入后的内存图（MZ—700模式）


$0000
监控程序
(ROM)
$1000
$D000
V-RAM
V-RAM
$E000
键和时钟口
$F000

中止


$0000
系统
$1000
BASIC
$D000

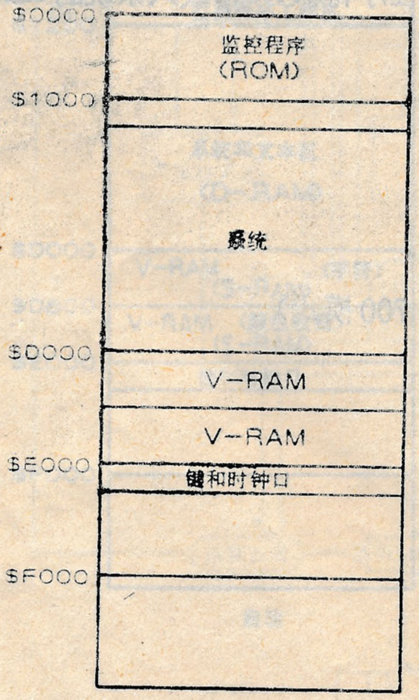
启动

- 以上是BASIC解释程序被载入后的内存图。
- 存储体转换系统用于存取来自V—RAM或从键和时钟口来的信息。

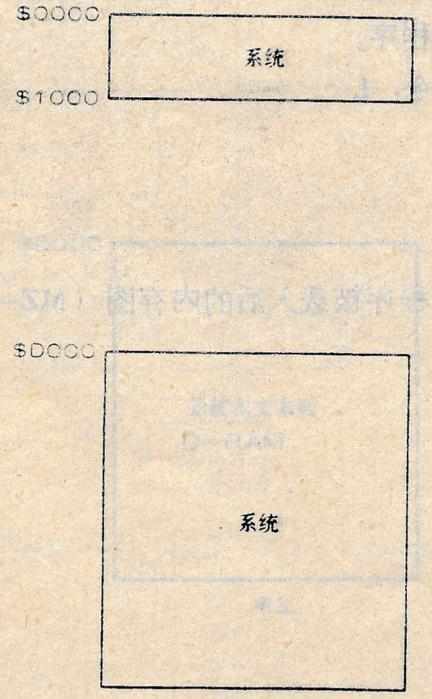
4. 2. 4

手动恢复后的内存图

计算机后面的复位开关被按下后，内存图如下所示：

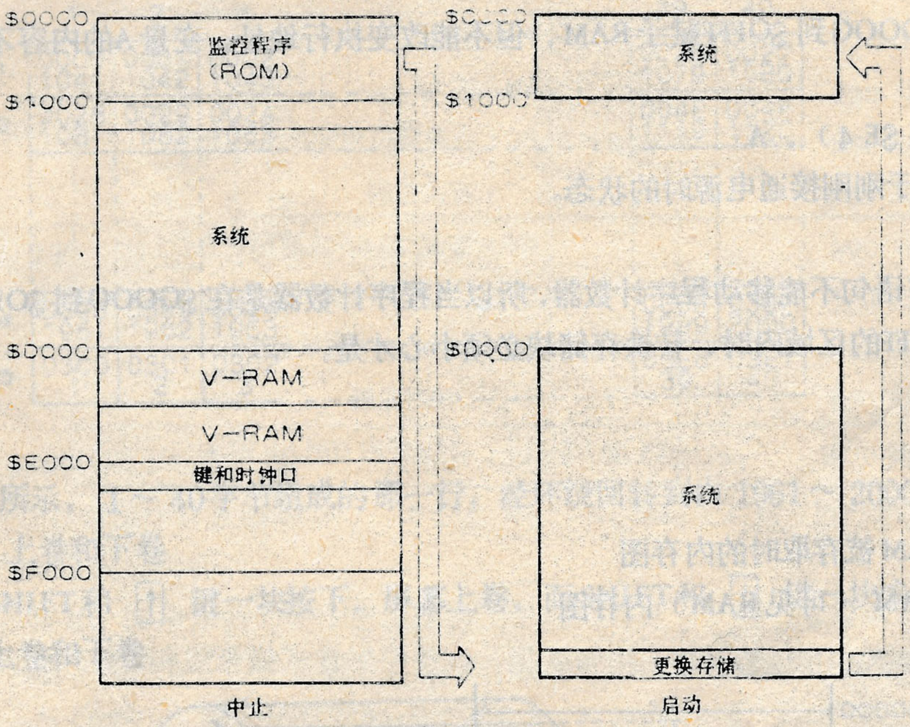

$0000
监控程序
(ROM)
$1000
系统
$D000
V-RAM
V-RAM
$E000
键和时钟口
$F000

启动


$0000
系统
$1000
$D000
系统

中止

复位开关与CTRL键一块按下后，内存图如下所示


$0000
监控程序
(ROM)
$1000
系统
$D000
V-RAM
V-RAM
$E000
键和时钟口
$F000
中止
$0000
系统
$1000
$D000
系统
更换存储
启动

- 当复位开关与CTRL键一块按下时,地址 $0000到$0FFF和 $D000到 $FFFF被赋予RAM。

- 当复位开关按下后键入#命令，计算机操作跟复位开关和 CTRL 键一块按下作用相同。

4·2、5

存储体的转换

a）输出数据到输入／输出（I／O）端口，选择的存储区如下所示。

I/O 口	$0000~$0FFF	$D000~$FFFF
$E0	系统区 (D-RAM)	
$E1		系统区 (D-RAM)
$E2	监控程序（ROM）	
$E3		V-RAM，键，时钟
$E4	监控程序（ROM）	V-RAM，键，时钟
$E5		禁止
$E6		这里是通过 $E5 来禁止返回前面状态

注意：

输出数据到I／O口 $E 4与按复位键执行的功能一样。

b）例子：

OUT （$EO），A

把地址 $OOOO到 $OFFF赋予RAM，但不能改变执行地址。变量A的内容不会影响结果。

OUT （$E 4），A

使内存置于刚刚接通电源时的状态。

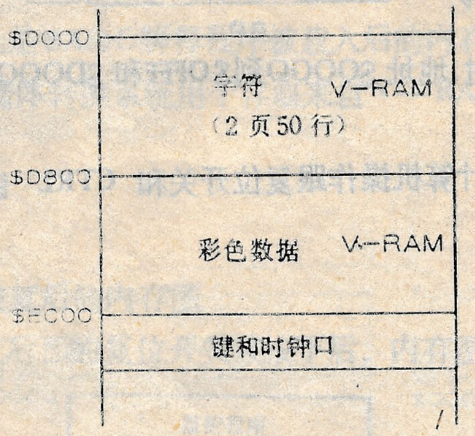
注意：

因为OUT语句不能移动程序计数器，所以当程序计数器是在 $OOOO到 $OFFF或 $DOOO到 $FFFF的区域内时，转换存储块必须小心才是。

4·2、6

当V—RAM被存取时的内存图

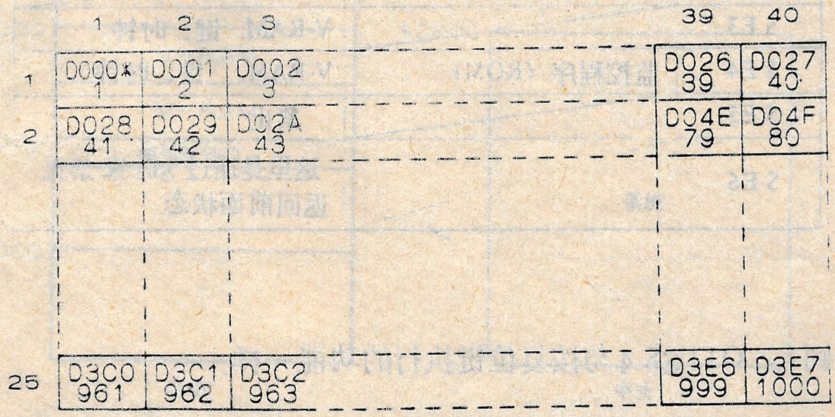
i）V—RAM （可见RAM）内存图


$D000
字符　V-RAM
（2 页50 行）
$D800
彩色数据　V-RAM
$E000
键和时钟口

ii）V—RAM的地址与屏幕的定位相一致。

MZ— 700有 2K 字节的 V—RAM区，可是在屏幕上一次只能显示这个区的 1K 字节。该区的显示可以通过卷起屏幕来改变。

a）复位键（或接通电源）后，马上显示的区域。


1
2
3
39
40
1
D000*
1
D001
2
D002
3
D026
39
D027
40
2
D028
41
D029
42
D02A
43
D04E
79
D04F
80
25
D3C0
961
D3C1
962
D3C2
963
D3E6
999
D3E7
1000

b）屏幕从V—RAM的端点上卷一行后的显示区：

	1	2	3		39	40
1	D000 1041	D001 1042	D002 1043		D026 1079	D027 1080
2	D028 1081	D029 1082	D02A 1083		D04E 1119	D04F 1120
24	D398 1961	D399 1962	D39A 1963		D3BE 1999	D3BF 2000
25	D3C0 1	D3C1 2	D3C2 3		D3E6 39	D3E7 40

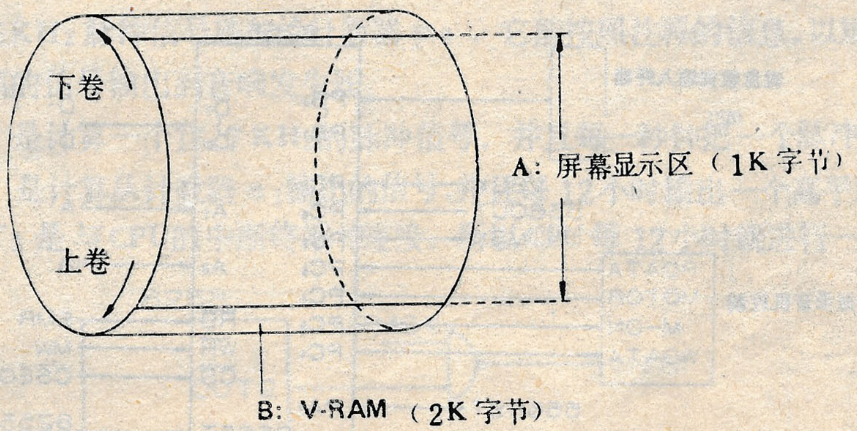
注意：

如上所示，1～40字节组成的那一行，经环绕回转到由1961～2000字节组成的行

iii）上卷和下卷

a）SHIFT和 ↑ 键一块按下，屏幕上卷，而SHIFT和 ↓ 键一块按下，则为下卷。

b）上卷和下卷



- 上图所示，在卷起时，在屏幕上显示的区域移动后可经过2K字节的RAM区。
- 下图所示，从V—RAM区的端点经移动可回转到RAM区的始点。
- 在卷起期间，屏幕上的光标不会移动。

第三节　I／O（$E 000—$E 008）输入／输出口的内存分配

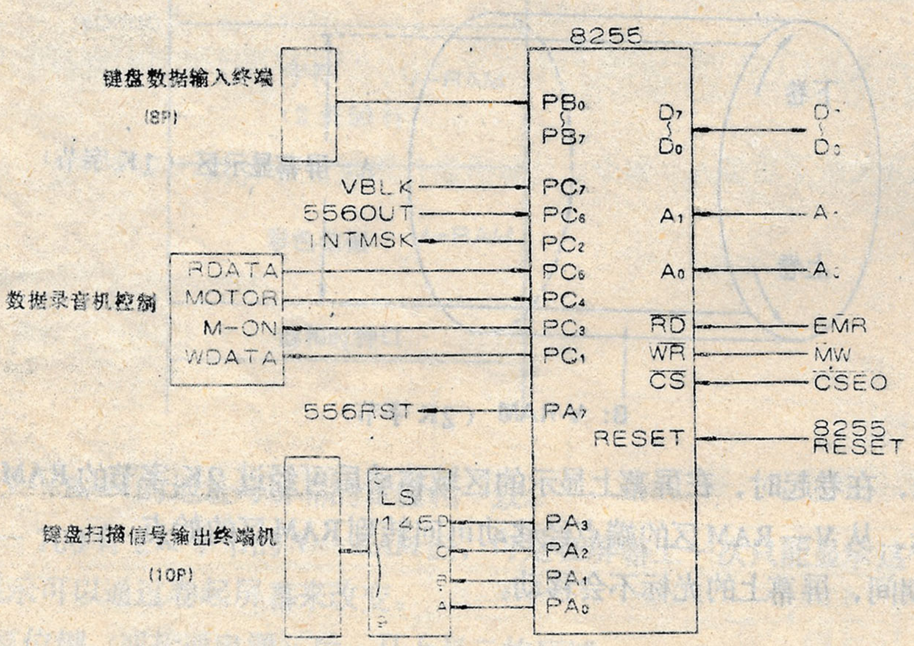
地址$E000到$E008被赋予8255可编程序外围接口以及8253可编程序时间间隔计时器和别的I／O控制ICs，这样使得各种I／O设备（包括音乐功能用的8253计数器＃0）的存取方式可用与内存相同的方法。I／O的内存分配如下所示：

CPU存储器地址	控制器	操　　作
$E 000	8255	PA：输出
$E 001		PB：输入
$E 002		PC：由设定位控制输入和输出
$E 003		方式控制
$E 004	8253	C 0：方式 3（方波速率发生器）
$E 005		C 1：方式 2（速率发生器）
$E 006		C 2：方式 0（终端计数器）
$E 007		方式控制
$E 008	LS 367等拍子，操纵杆和 HBLAK输入	

4·3、1

8255的信号系统

8255输出键盘扫描信号，输入按键数据，以及控制盒式磁带组和光标闪烁时限，


8255
键盘数据输入终端
(8P)
PB₀
PB₇
D₇
D₀
D
D
VBLK
556OUT
INTMSK
PC₇
PC₆
PC₂
PC₆
PC₄
PC₃
PC₁
A₁
A₀
A
A
数据录音机控制
RDATA
MOTOR
M-ON
WDATA
RD
WR
CS
EMR
MW
CSEO
556RST
PA₇
RESET
8255 RESET
键盘扫描信号输出终端机
(10P)
LS 145
D
C
B
A
PA₃
PA₂
PA₁
PA₀

端口	终端	I/O	活动状态	控制说明	信号名
PA ($E000)	PA_0	OUT	H	键盘扫描信号	
	PA_1		H		
	PA_2		H		
	PA_3		H		
	PA_7		L	恢复光标闪烁器	556 RST
PB ($E001)	PB_0	IN	L	键扫描数据输入信号	
	PB_1		L		
	PB_2		L		
	PB_3		L		
	PB_4		L		
	PB_5		L		
	PB_6		L		
	PB_7		L		

PC* ($E002)	PC$_1$	OUT	—	盒式磁带写数据	WDATA
	PC$_2$	OUT	L	禁止时钟中断	INTMSK
	PC$_3$	OUT	⎍	马达驱动信号	M－ON
	PC$_4$	IN	H	指明马达已接通	MOTOR
	PC$_5$	IN	—	盒式磁带读数据	RDATA
	PC$_6$	IN	—	光标闪烁定时器输入信号	556 OUT
	PC$_7$	IN	—	纵向闪烁信号	$\overline{\text{VBLK}}$

＊每个输出数据位可以独立地设置或重置。

4·3、2

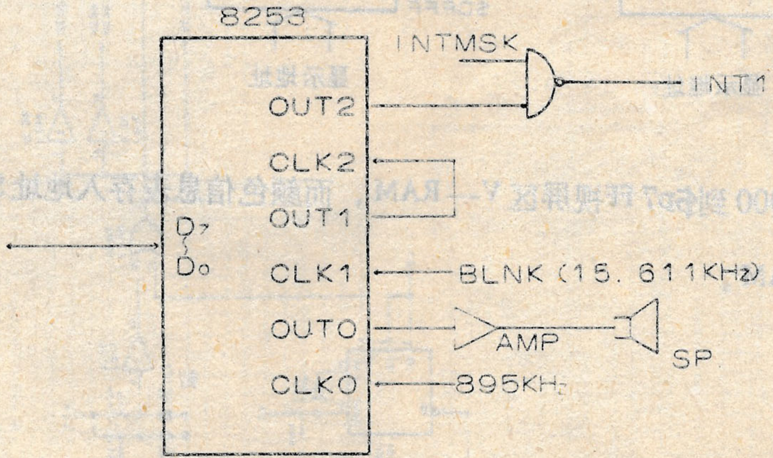
8253的信号系统

8253包含三个计数器即$\#_0$ ，$\#_1$ 和$\#_2$ 。计算器$\#_0$ 用于产生音响，而计数器$\#_1$ 和$\#_2$ 则用于内部时钟。

计算器$\#_0$ 被用来做为一个方波速率发生器（方式3）而计数器$\#_1$ 则被用来做一个速率发生器（方式2），计数器$\#_2$ 是在计数端点做为中断用（方式0）。

把一个895KH_2脉冲信号施加给计数器$\#_0$ ，它能按照注释的信息，以规定的值分离频率，这个分离的信号输出到音响发生器。

计数器$\#_1$ 是计算一个15.7KH_2的脉冲信号，并且每一秒钟把一个脉冲输出到OUT$_1$去。计数器$\#_2$ 是计算从计数器$\#_1$输出的信号，并且每12小时输出一个高平脉冲到OUT$_2$去，因为OUT$_2$是与CPU的中断终端相连接，所以CPU每12小时就进行一次中断。

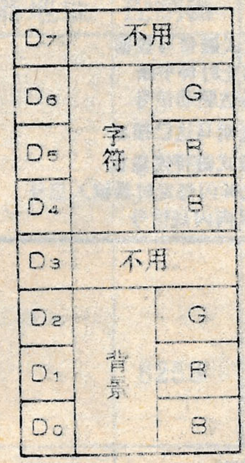

8253
INTMSK
INT1
OUT2
CLK2
OUT1
D7
~
D0
CLK1
BLNK (15. 611KHz)
OUT0
AMP
SP
CLK0
895KH2

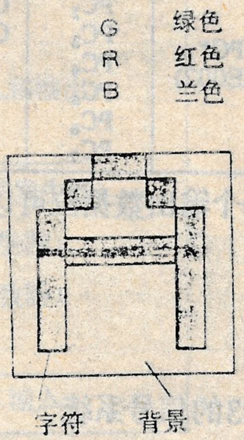
第四节　颜色V—RAM的信号系统

颜色V—RAM的信号系统

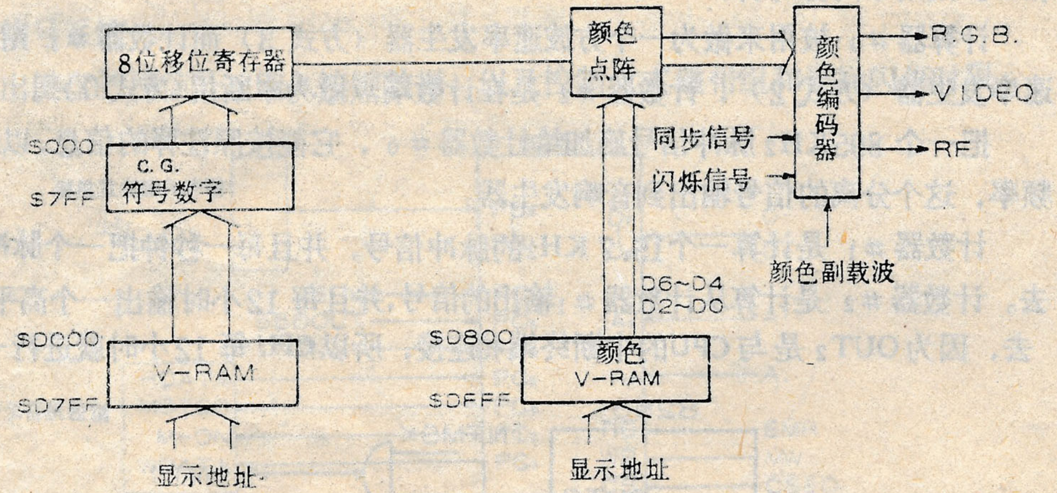
字符组由MZ—700的颜色信息控制，也就是说，一个字节的颜色信息表被赋予在屏幕上显示的每个字符

颜色信息表如下图所示：


D7
不用
D6
D5
D4
字符
G
R
B
D3
不用
D2
D1
D0
背景
G
R
B


G 绿色
R 红色
B 兰色
字符
背景

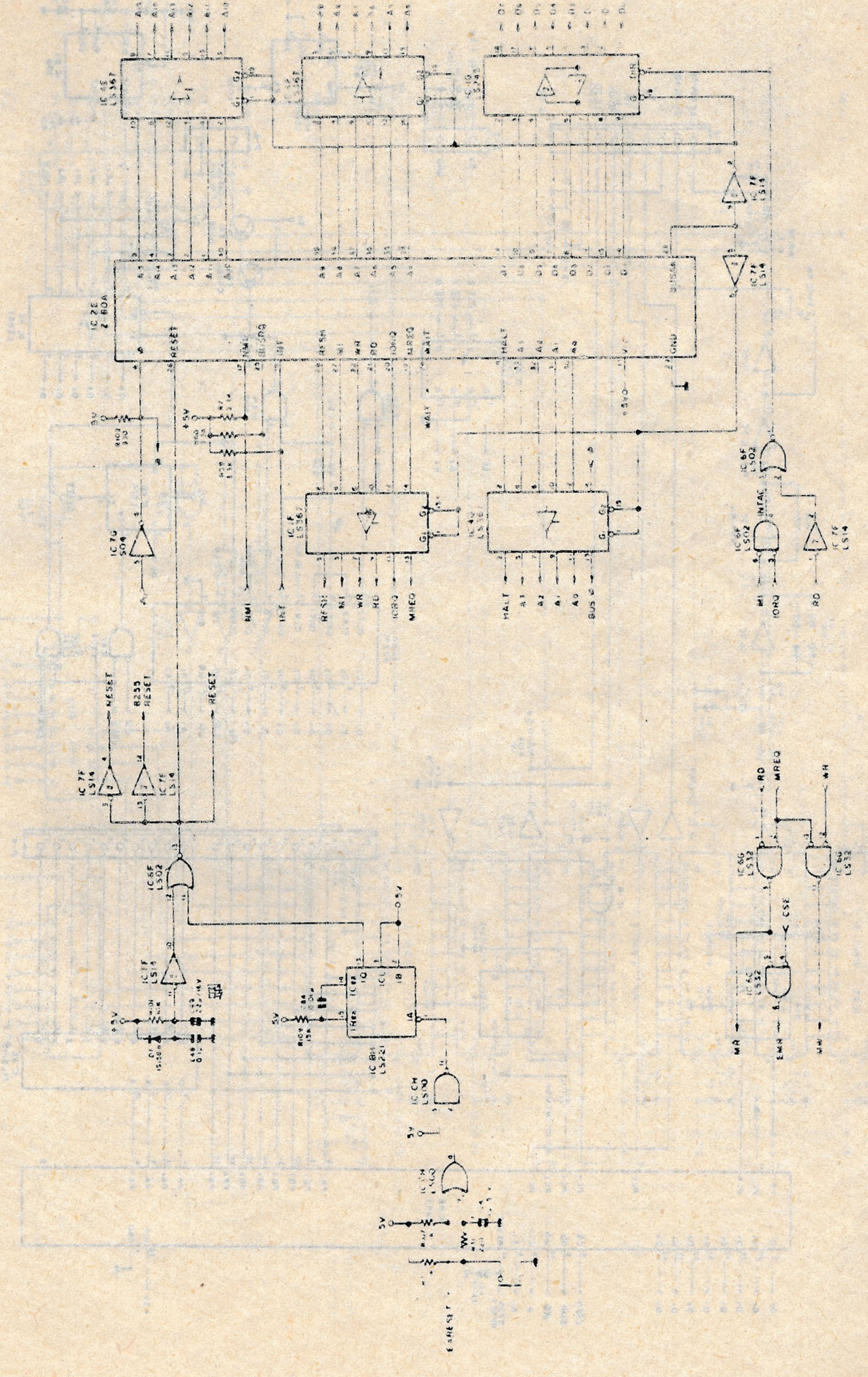
颜色信息表存取如下所示：


8位移位寄存器
$000
$7FF
C.G.
符号数字
$D000
$D7FF
V-RAM
显示地址
颜色
点阵
D6~D4
D2~D0
$D800
$DFFF
颜色
V-RAM
显示地址
颜色编码器
同步信号
闪烁信号
颜色副载波
R.G.B.
VIDEO
RF

字符显示存入地址$D 000到$D7 FF视屏区V—RAM，而颜色信息表存入地址$D 8000至$DFFF视屏区V—RAM。

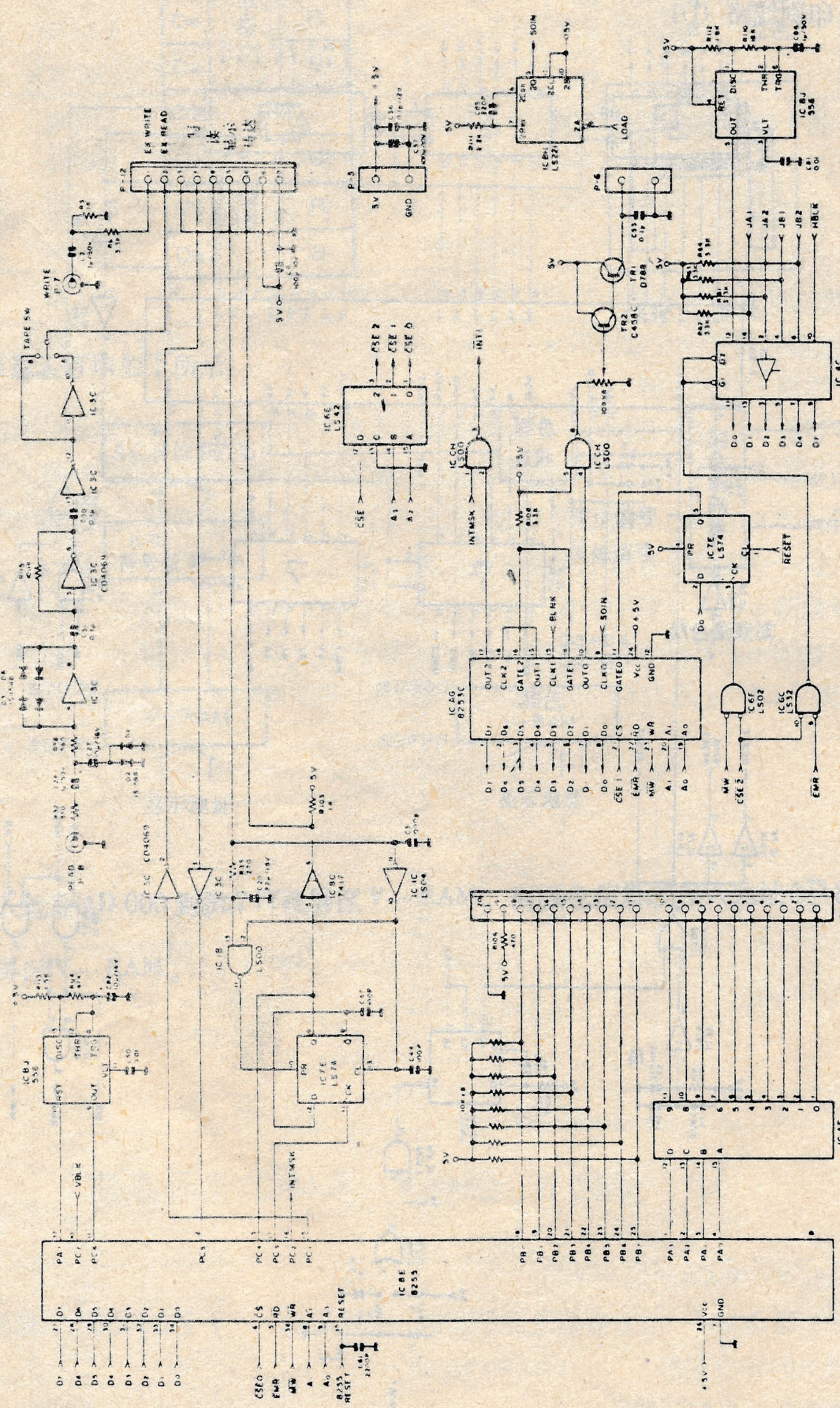
第五节 MZ—700 线路图

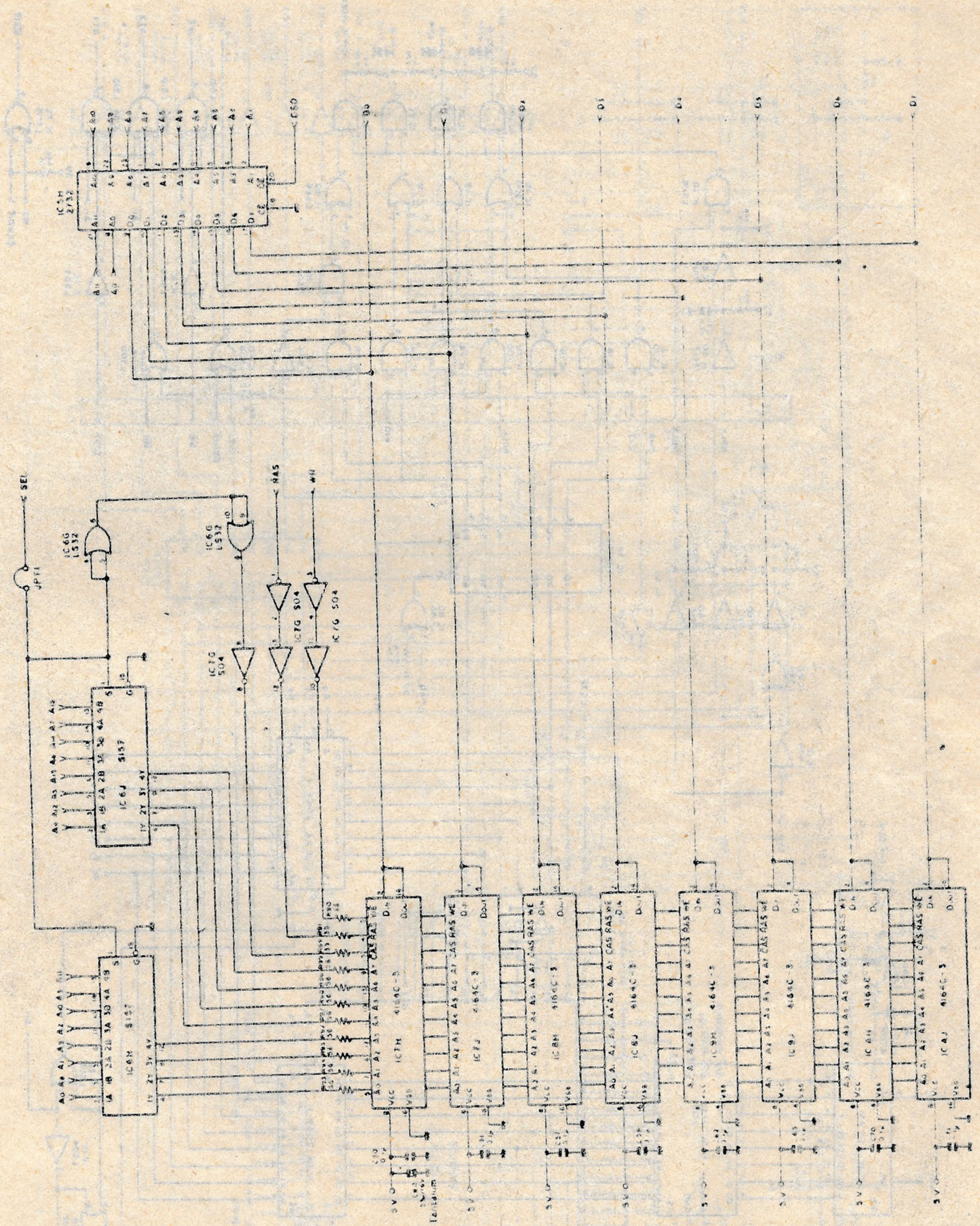
[CPU 印刷线路 (1)]



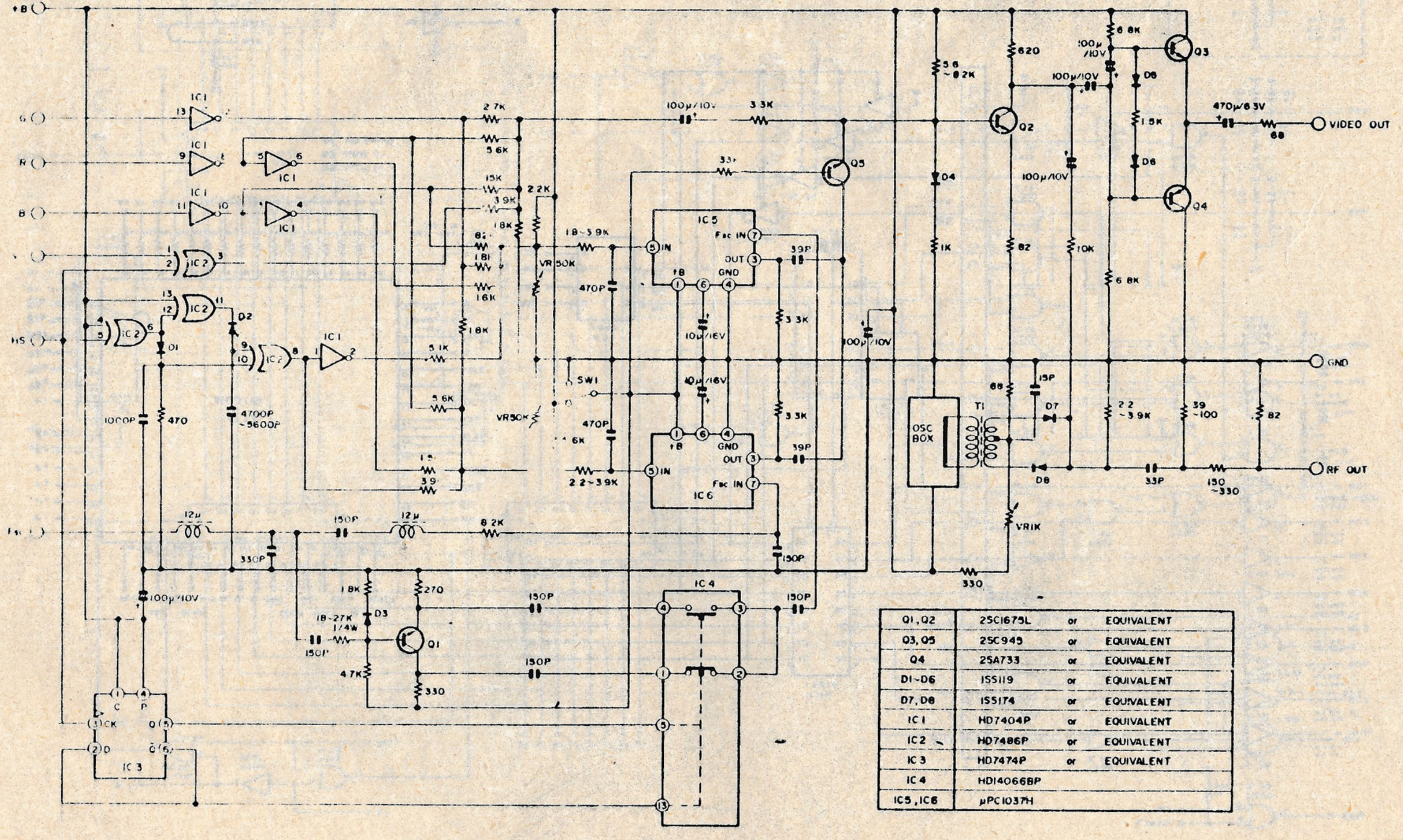
IC 2E
Z-80A
LS367
RESET
NMI
INT
RFSH
M1
WR
RD
IORQ
MREQ
WAIT
HALT
BUSAK
BUSRQ
GND
+5V
R255 RESET
LS14
LS02
LS04
LS32
LS221
LS10
INTAC
CSE
EMR

[CPU印刷线路 (2)]


P-12
EX WRITE
EX READ
P-3
5V
GND
P-6
P-7
WRITE
TAPE SW
IC 3C
CD4069
P-8
READ
IC 1B
LS00
IC 8C
7417
IC 1C
LS04
IC 8J
556
IC 7E
LS74
VBLK
INTMSK
IC 8E
8255
D7
D6
D5
D4
D3
D2
D1
D0
CSE0
EWR
MW
A1
A0
8255 RESET
RESET
+5V
VCC
GND
PA0
PC0
PC4
PC5
PC6
PC3
PC2
PC1
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PA1
PA2
PA3
IC AE
74LS145
P-4
IC 6E
LS42
PSE 2
CSE 1
CSE 0
CSE
A1
A2
IC CH
LS00
INT1
IC 6G
8253C
OUT2
CLK2
GATE2
OUT1
CLK1
GATE1
OUT0
CLK0
GATE0
BLNK
SOIN
LOAD
IC BH
LS221
TR1
D788
TR2
C458C
IC 7E
LS74
RESET
IC 6F
LS02
IC 6G
LS32
EMR
CSE 2
IC 4C
LS367
JA1
JA2
JB1
JB2
HBLK
IC 8J
556

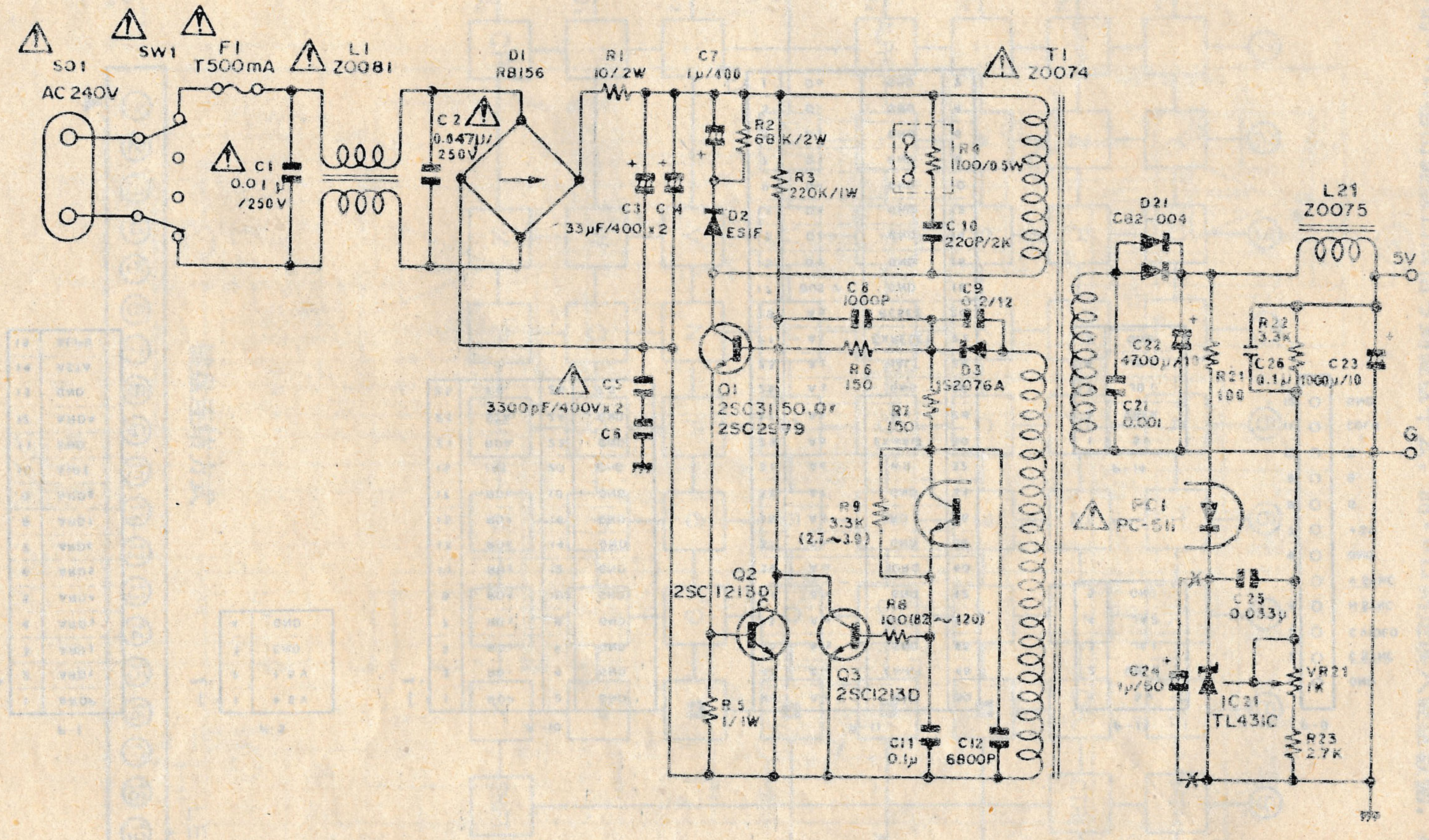

IC5M
2732
CSO
D0
D1
D2
D3
D4
D5
D6
D7
SEL
JPT1
IC6G
LS32
RAS
WR
IC7G
S04
IC6J
S157
IC6M
S157
IC7H
IC7J
IC8M
IC8J
IC9M
IC9J
IC8H
IC4J
4164C-3
5V

[彩色译码电路]


+B
G
R
B
HS
Fsc
VIDEO OUT
GND
RF OUT
IC1
IC2
IC3
IC4
IC5
IC6
Q1
Q2
Q3
Q4
Q5
D1
D2
D3
D4
D6
D7
D8
T1
OSC BOX
VR1K
VR50K
SW1
Fsc IN
OUT
GND
IN
+B
CK
D
C
P
Q
Q̄
470μ/6.3V
100μ/10V
10μ/16V
4700P ~5600P
1000P
330P
150P
470P
39P
33P
15P
12μ
1/4W
2.7K
5.6K
15K
3.9K
18K
22K
1.8~3.9K
2.2~3.9K
8.2K
5.1K
4.7K
1.8K
18~27K
6.8K
5.6~8.2K
620
82
68
330
270
470
1K
10K
1.5K
3.3K
39~100
150~330

Q1, Q2	2SC1675L	or	EQUIVALENT
Q3, Q5	2SC945	or	EQUIVALENT
Q4	2SA733	or	EQUIVALENT
D1~D6	1SS119	or	EQUIVALENT
D7, D8	1SS174	or	EQUIVALENT
IC1	HD7404P	or	EQUIVALENT
IC2	HD7486P	or	EQUIVALENT
IC3	HD7474P	or	EQUIVALENT
IC4	HD14066BP		
IC5, IC6	μPC1037H		

[电源]


SO1
AC 240V
SW1
F1
T500mA
L1
Z0081
C1
0.01μ
/250V
C2
0.047μ/
250V
D1
RB156
R1
10/2W
C3 C4
33μF/400×2
C7
1μ/400
R2
68K/2W
D2
ES1F
R3
220K/1W
R4
100/0.5W
C10
220P/2K
T1
Z0074
C5
3300pF/400V×2
C6
C8
1000P
C9
0.2/12
R6
150
D3
1S2076A
R7
150
Q1
2SC3150.Or
2SC2979
R9
3.3K
(2.7~3.9)
Q2
2SC1213D
Q3
2SC1213D
R8
100(82~120)
R5
1/1W
C11
0.1μ
C12
6800P
D21
C82-004
L21
Z0075
5V
C22
4700μ/10
C21
0.001
R21
100
R22
3.3K
C26
0.1μ
C23
1000μ/10
G
PC1
PC-511
C25
0.033μ
C24
1μ/50
IC21
TL431C
VR21
1K
R23
2.7K

[CPU终端印刷线路位置图]

P-1

1	ARDP
2	ARD_1
3	ARD_2
4	ARD_3
5	ARD_4
6	ARD_5
7	ARD_6
8	ARD_7
9	ARD_8
10	AIRT
11	GND
12	ARDA
13	GND
14	ASTA
15	ALPS

P-5

1	+5V
2	+5V
3	GND
4	GND

P-10

1	RDP	2	GND
3	RD_1	4	GND
5	RD_2	6	GND
7	RD_3	8	GND
9	RD_4	10	GND
11	RD_5	12	GND
13	RD_6	14	GND
15	RD_7	16	GND
17	RD_8	18	GND
19	IRT	20	GND
21	RDA	22	GND
23	$\overline{\text{STA}}$	24	GND
25	FG	26	FG

P-11

49	A_{15}	NMI	50
47	A_{14}	$\overline{\text{EXINT}}$	48
45	A_{13}	GND	46
43	A_{12}	$\overline{\text{MREQ}}$	44
41	A_{11}	GND	42
39	A_{10}	$\overline{\text{IORQ}}$	40
37	A_9	GND	38
35	A_8	$\overline{\text{RD}}$	36
33	A_7	GND	34
31	A_6	$\overline{\text{WR}}$	32
29	A_5	$\overline{\text{EXWAIT}}$	30
27	A_4	$\overline{\text{M1}}$	28
25	A_3	GND	26
23	A_2	$\overline{\text{HALT}}$	24
21	A_1	$\overline{\text{EXRESET}}$	22
19	A_0	RESET	20
17	BUS Ø	GND	18
15	D_7	GND	16
13	D_6	GND	14
11	D_5	GND	12
9	D_4	GND	10
7	D_3	GND	8
5	D_2	GND	6
3	D_1	GND	4
1	D_0	GND	2

P-13

1	5V
2	$\overline{\text{VBLK}}$
3	JA1
4	JA2
5	GND

P-14

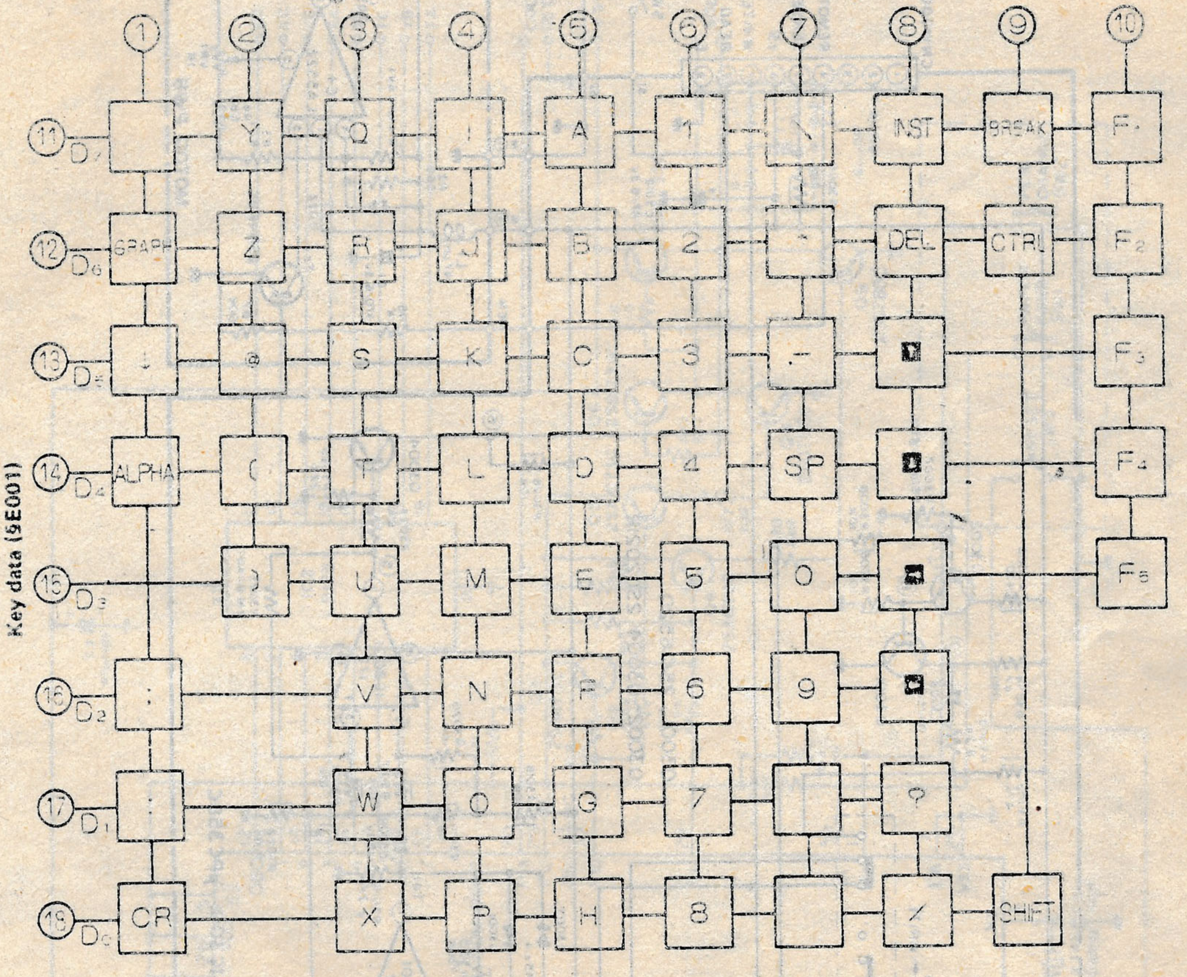
1	5V
2	$\overline{\text{VBLK}}$
3	JB1
4	JB2
5	GND

P-9

1	○	GND
2	○	$\overline{\text{C SYNC}}$
3	○	C VIDEO
4	○	$\overline{\text{H SYNC}}$
5	○	$\overline{\text{V SYNC}}$
6	○	GND
7	○	+5V
8	○	G
9	○	B
10	○	R
11	○	COLR
12	○	GND

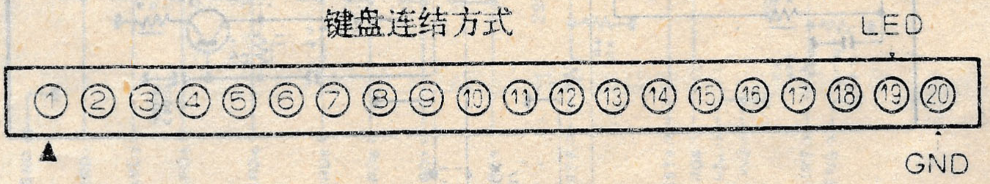
[键盘矩阵线路图]

从 8255 的 PA 口输出键盘扫描信号到键盘中去，而 PB 口则是读入键盘数据，其框图如下。

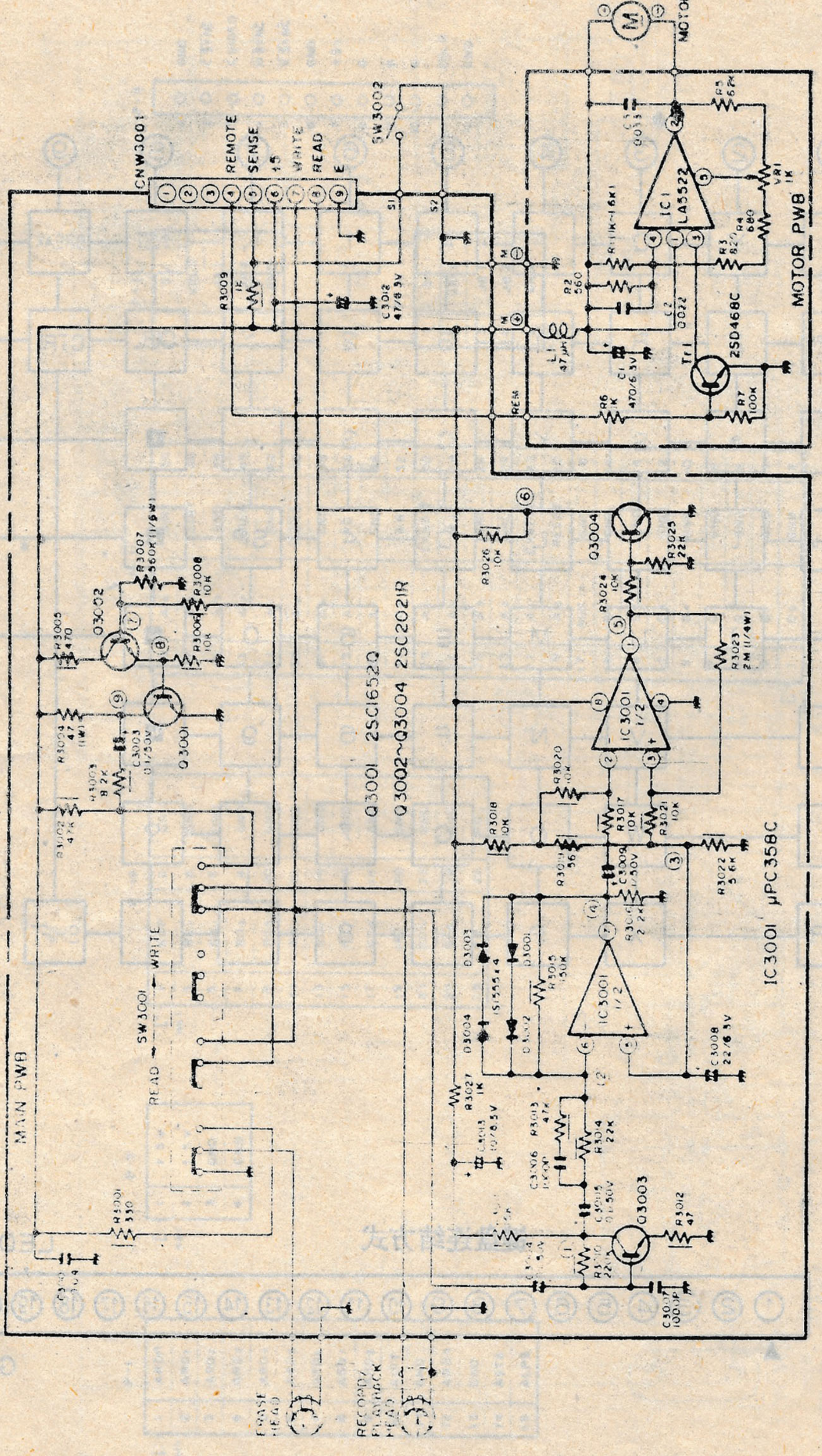

Key data (5E001)

	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩
⑪ D_7		Y	Q	I	A	1	\	INST	BREAK	F_1
⑫ D_6	GRAPH	Z	R	J	B	2	^	DEL	CTRL	F_2
⑬ D_5	↓	@	S	K	C	3	–	■		F_3
⑭ D_4	ALPHA	(	T	L	D	4	SP	■		F_4
⑮ D_3		)	U	M	E	5	0	■		F_5
⑯ D_2	;		V	N	F	6	9	■		
⑰ D_1	:		W	O	G	7		?		
⑱ D_0	CR		X	P	H	8		/	SHIFT	

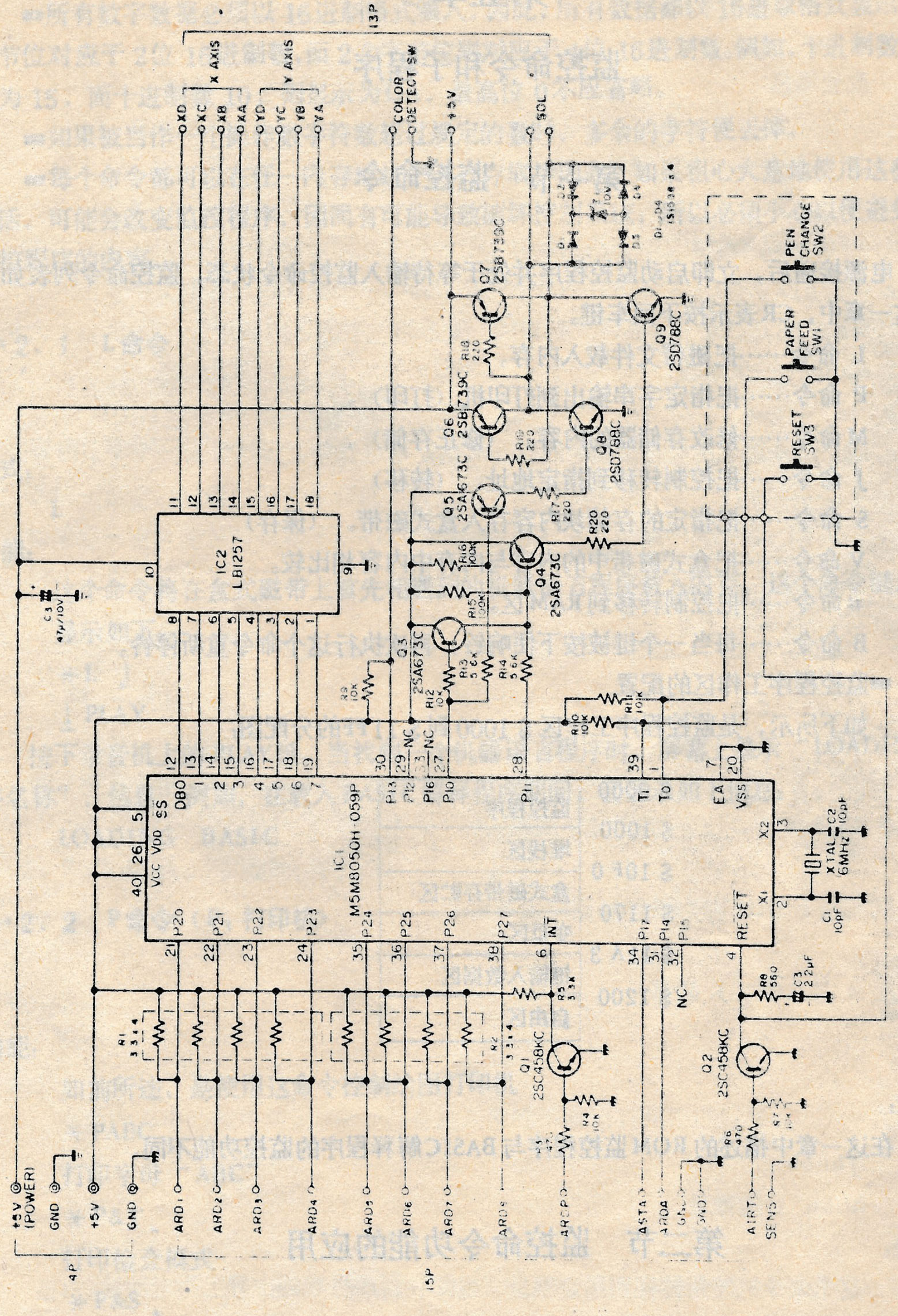
键盘连结方式


LED
① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳
GND

［数据录音机线路图］


CNW3001
REMOTE
SENSE
+5
WRITE
READ
E
SW3002
S1
S2
MOTOR
M
C3
.033
R5 6.2K
IC1 LA5522
VR1 1K
R1 11K~16K1
R2 560
R3 820
R4 680
C2 .022
L1 47μH
C1 470/6.3V
Tr1 2SD468C
R6 1K
R7 100K
REM
MOTOR PWB
R3009 1K
C3012 47/6.3V
Q3004
R3025 22K
R3026 10K
R3024 10K
R3023 2M (1/4W)
IC3001 1/2
R3020 10K
R3018 10K
R3017 10K
R3021 10K
R3019 56
C3009 .1/50V
R3022 5.6K
IC3001 μPC358C
Q3001 2SC1652Q
Q3002~Q3004 2SC2021R
R3007 560K (1/6W)
R3008 10K
R3006 10K
R3005 470
Q3002
Q3001
R3004 47 (1W)
C3003 .01/50V
R3003 8.2K
R3002 47K
SW3001
READ
WRITE
MAIN PWB
R3001 330
C3001 .01
D3003
D3004
D3001
D3002
1S1555×4
R3015 150K
R3016 2.2K
C3008 22/6.3V
R3027 1K
C3013 10/6.3V
R3013 47K
R3014 22K
C3006
C3015 .01/50V
Q3003
R3012 47
R3010 22K
C3014
C3017 100pF
ERASE HEAD
RECORD/PLAYBACK HEAD

[彩色绘图打印机线路图]


13P
X AXIS
Y AXIS
XD
XC
XB
XA
YD
YC
YB
YA
COLOR DETECT SW
+5V
SOL
IC2
LB1257
IC1
M5M8050H-059P
Q1 2SC458KC
Q2 2SC458KC
Q3 2SA673C
Q4 2SA673C
Q5 2SA673C
Q6 2SB739C
Q7 2SB739C
Q8 2SD768C
Q9 2SD768C
PEN CHANGE SW2
PAPER FEED SW1
RESET SW3
XTAL 6MHz
C1 10pF
C2 10pF
C3 2.2μF
C3 47μ/10V
R8 560
EA
VSS
RESET
X1
X2
DBO
VCC
VDD
SS
INT
+5V (POWER)
GND
+5V
GND
ARD1
ARD2
ARD3
ARD4
ARD5
ARD6
ARD7
ARD8
ARSPD
ASTB
ARDA
GND
GND
AIRTD
SENSO
NC
4P
15P

第五章

监控命令和子程序

第一节　监控命令

电源接通后，立即启动监控程序并处于等待输入监控命令状态。监控命令列表如下，在这一章中，CR表示按下回车键。

L 命令……把磁带文件载入内存。
P 命令……把指定字串输出到打印机（打印）。
M 命令……修改存储器的内容。（修正存储）。
J 命令……把控制转移到指定地址。（转移）
S 命令……把指定的存储块内容存入盒式磁带。（保存）
V 命令……把盒式磁带中的内容与内存中内容相比较。
#命令……把控制转移到RAM区。
B 命令……每当一个键被按下便响铃，再次执行这个命令重新停铃。

■监控程序工作区的配置

如下所示，是监控程序工作区 $ 1000到 $ 11FF的分配图：

$ 0000	监控程序
$ 1000	堆栈区
$ 10F0	盒式磁带存贮区
$ 1170	变量区
$ 11A3	键输入数据区
$ 1200	自由区

注意：

在这一章中描述的ROM监控程序与BASIC解释程序的监控功能不同。

第二节　监控命令功能的应用

这一节讨论每个监控命令的功能和应用

■当按下CR键后，命令被执行。字符必须按正确的次序输入，如果命令串中含有非法字符（如空格），那么监控程序会拒绝执行这个命令。

■所有数字数据必须以16进制格式输入，因此，所有数据都以16进取格式显示。一个字节位对应于2位16进制数，而2个字节位则对应于4位16进制数。例如，十进制数21显示为15，而十进制数10，则显示为OA，最高位0不应省略。

■如果被当作一个操作数字符数超过规定的数时，多余的字符便丢掉。

■每个命令都可以在任一内存地址中进行存取，因此，如果粗心大意地使用这些命令的话，可能会改变监控程序，因而有可能导致破坏控制系统。所以必须小心以便避免改变监控程序的内容。

5·2、1　L命令

格式：

L

功能：

这个命令将在盒式磁带上首先相遇到的机器语言程序载入内存。这个命令键入后，显示如下：

*L ↵

↓ PLAY

按下录音机上的PLAY键。当找到一个机器语言程序时，屏幕上显示“LOADING程序名称”，信息。例如，在载入BASIC解释程序期间，会显示如下信息：

LOADING　BASIC

5·2、2　P命令（P；打印机）

功能：

如前所述，是使用这命令控制绘图打印机

*PABC ▴

打印字母“ABC”

*P&T ▴

打印检查模式

*P&S ▴

置每行打印80个字符。

```
*P&L ↲
```

置每行打印40个字符

```
*P&G ↲
```

把打印机换为绘图方式

```
*P&C ↲
```

改变笔色

5·2、3　M命令（M：修改内容）

格式：

Mhhhh

hhhh……串地址

功能：

这个命令用于改变存储器的内容，从指定地址开始，每次改变一个字节内容。

```
*MC000↲
C000  00  FF
C001  00  FF
C002  00  FF
C003  00  FF
C004  00      SHIFT + BREAK   (即两键同时按下)
*MC010↲
C010  00  88
C011  00  88
C012  00  88
C013  00  88
C014  00      SHIFT + BREAK
*
```

同时按下SHIFT和BREAK键终止M命令。

5·2、4　J命令（J：转移）

格式：

Jhhhh

hhhh　目的地址

功能：

这个命令把控制转移到指定地址，也即在程序计数器中置上指定地址。

*J 1200 ↲……转移到地址$1200

5·2、5　S命令（S：存带）

格式：

Shhhhh′h′h′h′h″h″h″h″

hhhh　始地址

h′h′h′h′　末地址

h″h″h″h″　执行地址

在执行中，这个命令提醒你输入文件名字，然后把内存地址hhhh到h′h′h′h′的内容，存入指定的文件名之后的磁带中。假定有一个在内存地址为$6000到$60A3，执行地址为$6050的一个机器语言程序，以名字“MFILE”存带时，该命令的键入如下所示：

```
*S600060A36050↵
FILENAME? MFILE↵
⤓ RECORD.PLAY
```

确信一盒空白磁带已装在录音机上后，按下RECORD键。

如果磁带盒的防护片被卸了，RECORD键就按不下，应换一盒磁带。

这个命令只能用于保存机器语言程序。

```
WRITING MFILE
OK!
※ →光标
```

为了中止录制，可把SHIFT和BREAK键一块按下直到出现指示符“*”为止。

5·2、6　V命令（V；校对）

格式：

V

功能：

是将内存中的程序与用S命令存在磁带中的机器语言程序进行比较。

```
* V
⤓ PLAY
OK
```

当显示提示符“⤓ PLAY”时，按下PLAY键，就开始读磁带中的文件。当磁带中文件的内容与内存中原程序的内容完全相同时，会显示“OK”字样，否则将显示信息“check

SUM ERROR”（检查和错）

我们建议，在用S命令录制一个程序后，最好是执行这个命令。

5·2、7 #命令

格式:

#

功能:

按下RESET开关后，执行这个命令与同时按下RESET开关和CTRL键的效果相同。

＊#↲

5·2、8 B命令（B；铃）

格式:

B

功能：＊B↲

一旦执行这个命令后，每当按下一个键，铃就会响。再次执行它将停止响铃。

第三节　监控程序的子程序

监控程序1Z—013A提供了下列子程序（见表），在这里每个子程序的名字都象征性地表达了相应的子程序的功能。这些子程序可由用户程调用。

当控制返回到调用的程序时，被保留的寄存器的内容将被重新恢复。别的寄存器的内容可能由于执行子程序而被更改。

名字和子程序的进入点（16进制）	功　　能	被保留的寄存器
CALL LETNL （0006）	移动光标到下一行的起始位	除AF以外
CALL PRINTS （000C）	在光标位置显示一个空格	除AF以外
CALL PRINTS （0012）	在光标位置上，显示存在ACC中的ASCII代码的对应字符，关于ASCII代码请看附录A 1。当代码OD（回车）或11到16（光标控制代码）被输入时，没有字符显示，但仍会执行行相应的功能（对OD是回车，而11到16是移动光标）。	除AF外

CALL MSG （0015）	在光标位置显示一个信息，在调用这个子程序之前，存储该信息的内存区的起始地址必须存入DE寄存器中，而该信息的末尾也必须有一个回车代码(OD)该回车码不会被执行，倘使任何一个控制光标的代码（11～16）被包含在信息中的话，光标会被移动。	所有寄存器
CALL BELL （003E）	短促的高音A（大约880Hz）	其它AF
CALL MELDY （0030）	按照DE寄存器所指示的地址开始的内存区中所保存的音乐数据，演奏音乐。音乐数据必须与BASIC的音乐语句所使用的格式相同且必须用OD或C8结束 演奏完后，控制返回到调用程序而特征位C则置为0，当演奏是用BREAK键中止时，控制返回而特征位被置为1。	除AF以外
CALL XTEMP （0041）	把按照拍子数据设置的音乐拍子存入累加器（ACC）ACC←01最低速。ACC←04中速，ACC←07最高速。 注：在累加器中的这个数据不是ASCII代码，而是相应于1～7的二进制码。	所有寄存器
CALL MSTA （0044）	产生一个规定频率的连续音响，这个频率由如下方程给出： freq＝895KHZ／nn'。 这里的nn'是存放在地址11A1和11A2中（n存在11A2，n'存在11A1）的一个二个字节的数。	BC和DE
CALL MSTP （0047）	停止由CALL MSTA子程序产生的音响	除AF以外
CALL TIMST （0033）	设置和启动内部时钟。在调用这个子程序以前寄存器必须做如下设置：ACC←0（AM），ACC←1（PM） DE←以秒为单位表示的时间(4位16进制数）	除AF以外
CALL TIMRD （003B）	读出内部时钟然后恢复时间如下 ACC←0（AM），ACC←1（PM） DE←以秒表示时间的一个4位16进制数。	除AF和DE以外
CALL BRKEY （001E）	检查SHIFT和BREAK两个键是否按下，当两个键被同时按下时，就置特征位Z，否则就使它复位。	除AF以外
CALL GETL （0003）	从键盘中读入一行数据并且把它存放在DE寄存器所指示的地址开始的内存区中。当按下RETURN键时，这个子程序就停止读数据，而挂上一个回车代码（OD）就结束读数据，一行最多可输入80个字符（包括回车代码）键入的字符反射	所有寄存器

	回显示屏，且光标控制代码可以被包含在该行内 当SHIFT和BREAK被同时按下时，BRAEK 代码被存在DE寄存器所指示地址中，而回车代码则被存在其后的地址单元内	
CALL GETKY （001B）	从键盘读入一个字符代码（ASCII）如果没有键被按下，控制返回到调用程序，且把累加器（ACC）置00 既没有提供避免由于键振动而使数据读错的设施，也没有把键入的字符反射回显示屏 当任何一个特殊键（如DEL或CR）被按下时，这个子程序就把一个代码送返累加器（ACC），如下所示这个代码是不同于相应的ASCII代码的 在这里，显示代码习惯于使用存在字符发生器中的字符地址从而别区于ASCII代码	除AF以外

	特殊键	存放在ACC中的代码	显示代码
用GETKY 读特殊键	DEL	60	C7
	INST	61	C8
	ALPHA	62	C9
	BREAK	64	CB
	CR	66	CD
	↓	11	C1
	↑	12	C2
	→	13	C3
	←	14	C4
	HOME	15	C5
	CLR	16	C6

名字和子程序进入点（16进制）	功能	被保留的寄存器
CALL ASC （03DA）	以相应的16进制数设置的ASCII字符，用ACC中低4位表示	除AF以外
CALL HEX （03F9）	把存放在ACC中的8位数转换成16进制数（假设这个数是一个ASCII字符）然后把这个16进制数存入ACC的最低4位 当一个16进制数存放在ACC中时，特征位C置0，否则置1	除AF以外
CALL HLHEX （0410）	把有4个ASCII字符的字串转换成16进制数，然后把它存入HL寄存器中，调用和返回条件如下所示： DE←含有ASCII字符串的内存区起始地址 （例如 "3" "1" "A" "5" ↑———DE CALL HLNEX CF＝0 HL←16进制数，例如： （HL＝31A5H）CF＝1 ACC的内容未确定	除AF HL以外

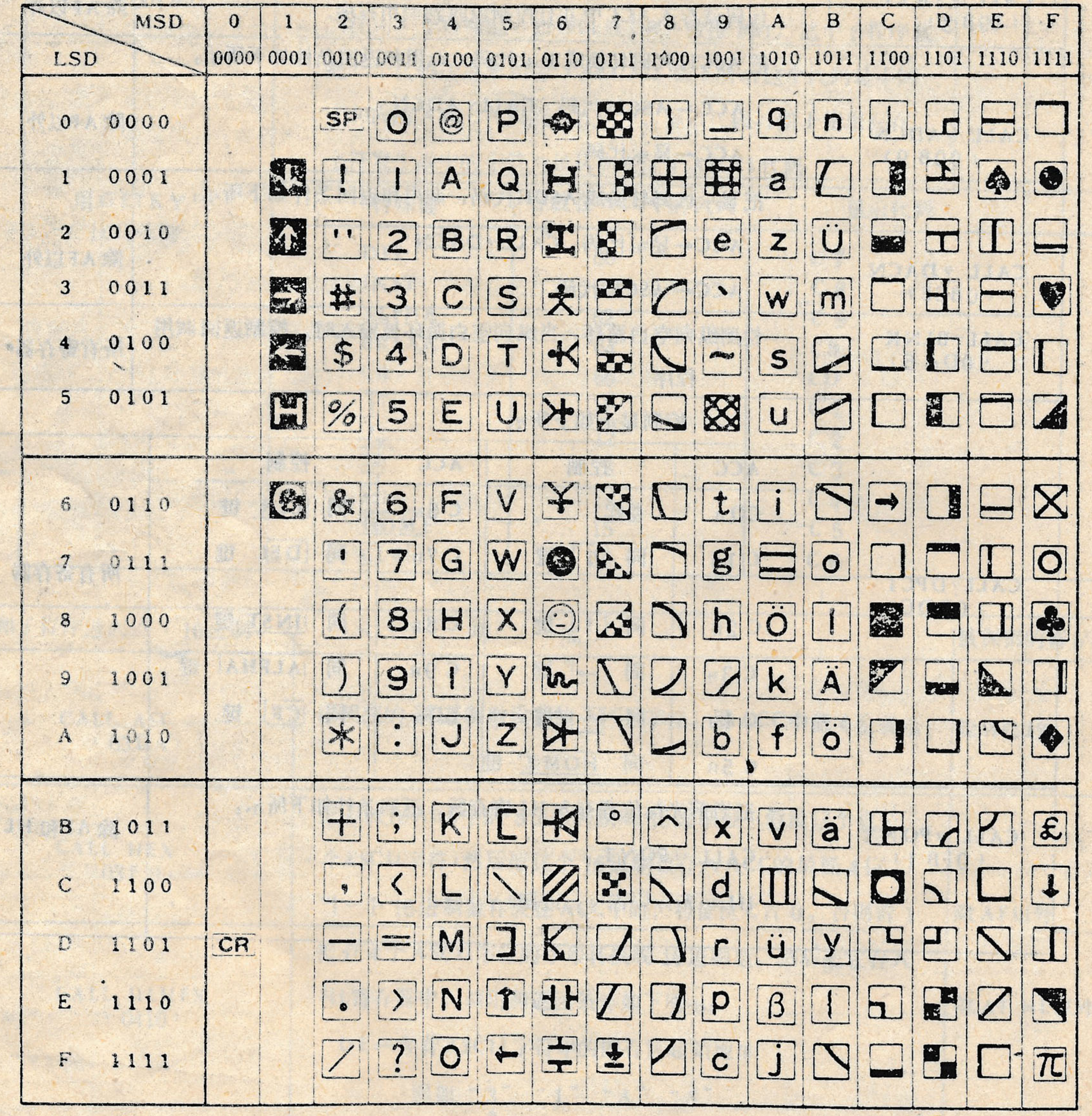
<table>
<tr><td>CALL 2HEX
（041F）</td><td>把含有2个ASCII字符的字串 换成一个16进制数，并把它置于ACC中 调用和返回条件如下所示：
DE←容纳ASCII字符串的内存区的首地址
（如 "3" "A"
└—DE CF= 0 ACC← 16进制数（ACC= 3AH）
CALL 2HEX CF= 1：ACC的内容未确定</td><td>除AF和DE以外</td></tr>
<tr><td>CALL ??KEY
（09B3）</td><td>光标闪烁以提示按键输入 当一个键被按下时，相应的显示代码就存入ACC中并且控制返回到调用程序</td><td>除AF以外</td></tr>
<tr><td>CALL ?ADCN
（00B9）</td><td>把ASCII代码转换成显示代码 调用和返回条件如下所示：
ACC←ASCII代码 CALL?ADCN
ACC←显示代码</td><td>除AF以外</td></tr>
<tr><td>CALL ?DACN
（0BCE）</td><td>把显示代码转换成ASCII代码，调用和返回条件如下所示：
ACC←显示代码 CALL?DACN
ACC←ASCII代码</td><td>除AF以外</td></tr>
<tr><td>CALL?BLNK
（0DA6）</td><td>检测纵向空白循环，当纵向空白循环被输入时，控制返回调用程序</td><td>所有寄存器</td></tr>
<tr><td>CALL?DPCT
（0DDC）</td><td>控制显示如下所示
<table>
<tr><th>ACC</th><th>控制</th><th>ACC</th><th>控制</th></tr>
<tr><td>C0$_H$</td><td>卷起</td><td>C6$_H$</td><td>同 CLR 键</td></tr>
<tr><td>C1$_H$</td><td>同 ↓ 键</td><td>C7$_H$</td><td>同 DEL 键</td></tr>
<tr><td>C2$_H$</td><td>同 ↑ 键</td><td>C8$_H$</td><td>同 INST 键</td></tr>
<tr><td>C3$_H$</td><td>同 → 键</td><td>C9$_H$</td><td>同 ALPHA 键</td></tr>
<tr><td>C4$_H$</td><td>同 ← 键</td><td>CD$_H$</td><td>同 CR 键</td></tr>
<tr><td>C5$_H$</td><td>同 HOME 键</td><td></td><td></td></tr>
</table></td><td>所有寄存器</td></tr>
<tr><td>CALL ?PONT
（0FB1）</td><td>把当前光标位置存入HL寄存器，返回条件如下所示：
CALL ?PONT
HL←光标位置（二进制）</td><td>除AF和HL外</td></tr>
</table>

附　　录

A. 1　代码表

■ASCII 代码表

MSD是最高有效位的缩写，代表每个代码的高 4 位。LSD是最低有效位的缩写，代表每个代码的低 4 位。代码 11H到 16H是光标控制码。例如把 15H赋予ACC，执行CALL PRINT（一个监控子程序）将使光标回到左上角。（不显示“H”）



MSD LSD	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0　0000			SP	0	@	P	◈	▦	│	_	q	n	│	▗	─	▢
1　0001		↓	!	1	A	Q	H	▚	⊞	▦	a	⧄	▐	╧	♠	●
2　0010		↑	"	2	B	R	⌶	▞	◜	e	z	Ü	▄	╪	│	▁
3　0011		→	#	3	C	S	🯅	▀	◜	`	w	m	▔	╫	═	♥
4　0100		←	$	4	D	T	🯆	▄	◟	~	s	⧄	▁	▌	═	▏
5　0101		H	%	5	E	U	🯇	◩	◟	▩	u	⧄	▏	▌	▔	◢
6　0110		↺	&	6	F	V	¥	◪	◟	t	i	⧅	→	▐	▁	☒
7　0111			'	7	G	W	●	◩	◝	g	≡	o	▕	▔	‖	○
8　1000			(	8	H	X	☺	◪	◝	h	ö	l	█	▀	‖	♣
9　1001			)	9	I	Y	∿	⧅	◞	⧄	k	Ä	◤	▄	◣	▕
A　1010			*	:	J	Z	🯈	⧅	◞	b	f	ö	▐	▢	◝	♦
B　1011			+	;	K	[	🯉	°	^	x	v	ä	╞	◞	◜	£
C　1100			,	<	L	\	▨	⁙	⧅	d	▥	⧅	◙	◟	▢	↓
D　1101	CR		-	=	M	]	K	⧄	◝	r	ü	y	┗	┛	⧅	‖
E　1110			.	>	N	↑	╫	⧄	◞	p	β	{	┓	▚	⧄	◥
F　1111			/	?	O	←	÷	⤓	⧄	c	j	⧅	▂	▚	▢	π

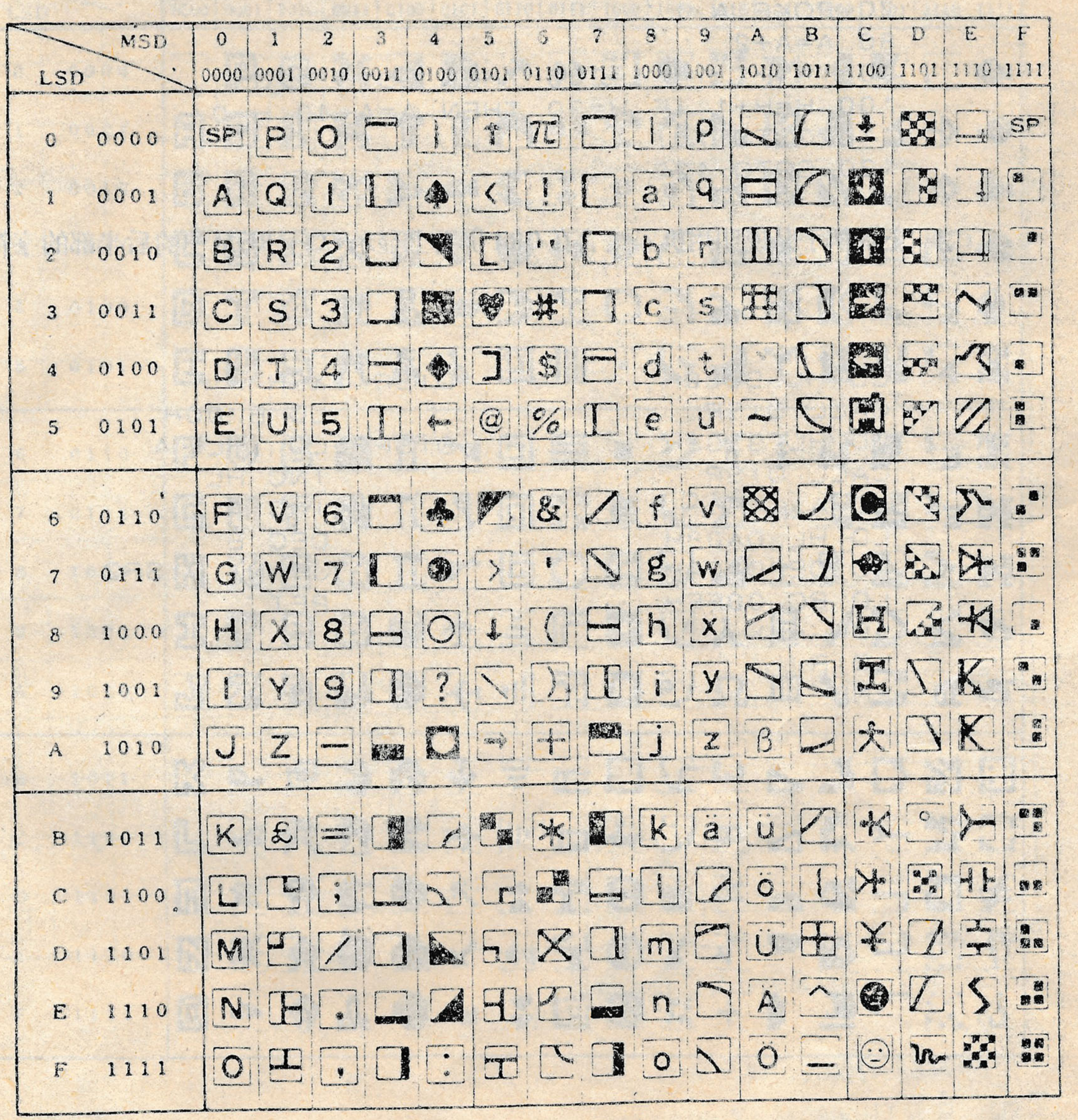
■显示代码表：

显示代码作为地址字符图形存入字符发生器中。为了显示字符，必须把这些代码送到视频随机存取存贮器（V—RAM）。

监控子程序PRINT（0012H）和MSG（0015H）用来把ASCII代码转换成显示代码.

然后把它们传送到视频随机存取存贮器以指出光标位置。

代码C1H到C6H用以控制光标



MSD / LSD	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	SP	P	0		│	↑	π			p						SP
1 0001	A	Q	1		♠	<	!		a	q						
2 0010	B	R	2			[	"		b	r						
3 0011	C	S	3			♥			c	s						
4 0100	D	T	4		♦	]	$		d	t						
5 0101	E	U	5		←	@	%		e	u	~					
6 0110	F	V	6		♣		&		f	v						
7 0111	G	W	7		●	>	'		g	w						
8 1000	H	X	8		○	↓	(		h	x						
9 1001	I	Y	9		?		)		i	y						
A 1010	J	Z	—			→	+		j	z	β					
B 1011	K	£	=				*		k	ä	ü					
C 1100	L		;						│		ö					
D 1101	M		/						m		ü		¥		÷	
E 1110	N		.						n		Ä	^				
F 1111	O		,		:				o		Ö	_	☺			

在前页中的字符模式被包含在计算机制图只读存贮器（CG—ROM）的 4K 个字节前半部，而在下页上的制图模式则被放在它的 4K 个字节的后半部。然而在CG—ROM的后半部的字符模式却不能被BASIC支持，且不能从键盘直接输入，如下面的程序例子所示，尽管可用POKE语句显示它们，但不能在任何打印机（内部打印机或外部打印机）上打印输出。

例如：

（1）下面的程序例子是在CRT屏幕上显示 CG—ROM的后半部的字符模式，

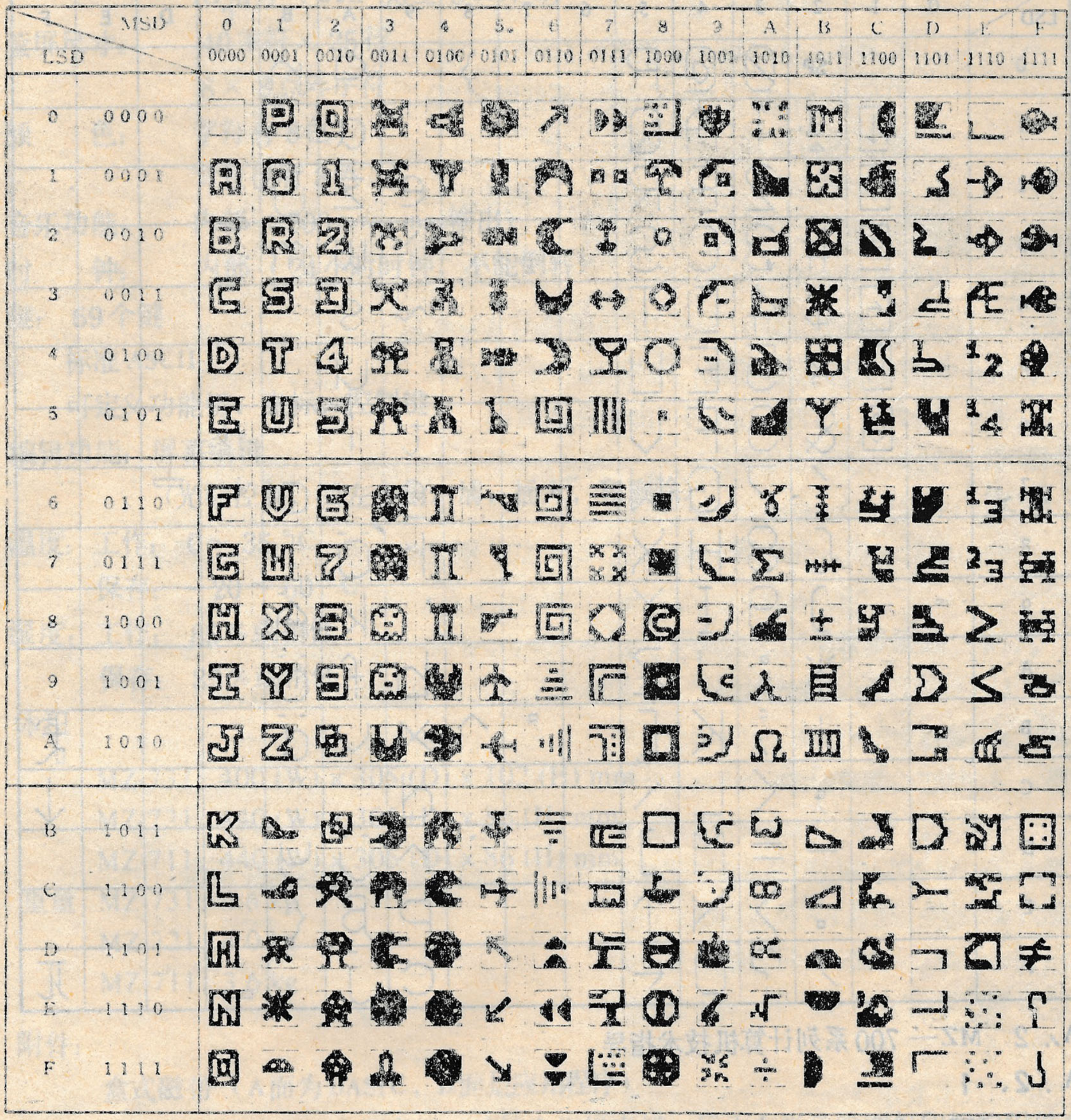
```
10 COLOR, , 7, 0
20 PRINT"■";
```

```
30  FOR J=55296 TO 56296  ———→ 55296 = $D800
40  POKE J. 240  ————————————→ Specifies the second 2K-byte
50  NEXT J                     half of CG-ROM. 240 = $F0
60  A=53248:I=O:H=O  ————————→ 53248 = $D000
70  POKE A. I
80  A=A+2
90  I=I+1:IF I=256 THEN GOTO 120
100 H=H+1:IF H=20 THEN A=A+40:H=0
110 GOTO 70
120 GOTO 120
```

（2）下面例子是说明用机器语言在CRT屏幕上显示CG—ROM中的后半部的字符模

```
LD  HL. D000H              DISP  :XOR A
CALL DISP                         LD B. 00H
LD  HL. D208H              DISP2:LD  (HL). A
CALL DISP                         INC HL
LD  A. F1H                        INC A
LD  HL. DA08H                     DEC B
LD  DE. DA09H                     JP NZ. DISP2
LD  BC. 00FFH                     RET
LD   (HL). A
LDIR
END
```

■MZ－700的显示代码表


MSD
LSD
0 0000
1 0001
2 0010
3 0011
4 0100
5 0101
6 0110
7 0111
8 1000
9 1001
A 1010
B 1011
C 1100
D 1101
E 1110
F 1111

■关于彩色绘图打印机的ASCII代码表

绘图字符而不是上面所示的不可打印字符，且可以用不同的色笔打印对应的十六进制代码。

MSD LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P			}		q	n				
1		⍗	!	1	A	Q					a					
2		⍐	"	2	B	R				e	z	Ü				
3		⍈	#	3	C	S				`	w	m				
4		⍇	$	4	D	T				~	s					
5		🄷	%	5	E	U					u					
6		🄲	&	6	F	V				t	i		→			
7			'	7	G	W				g		o		▔		
8			(	8	H	X				h	Ö	l				
9			)	9	I	Y					k	Ä				
A			*	:	J	Z				b	f	ö				
B			+	;	K	[		°	^	x	v	ä				£
C			,	<	L	\				d						↓
D			-	=	M	]				r	ü	y				
E			.	>	N	↑				p	ß	{				
F			/	?	O	←				c	j		▁			π

A. 2　MZ—700系列计算机技术指导

A. 2、1

MZ—700

CPU：夏普LH0080A（Z 80A）

时钟：3.5MH$_2$

内存：ROM　4K 字节（ROM）
2K 字节（字符发生器）
RAM　64K 字节（程序区）
4K 字节（视频 RAM）

视频输出：
PAL系统
RGB信号

组合　信号（B／W）

RF信号（UHF 36±3CH，B／W）

萤屏尺寸：　40字符×25行

8×8点阵字符

颜　　色：　字符有8色

背景也有8色

音乐功能：　内部（500mw　max输出）

时　　钟：　内部（24小时时钟，不能倒转）

键：69个键

标准ASCII

可定义功能键，光标，控制键

编辑功能：屏幕编辑

（光标控制，回左上角，清，插入，和删除）

温度：工作：0～35°C

保存：－20～60°C

湿度：工作：85％或稍低

保存：85％或稍低

体积

MZ-731：400（W）×305（D）×102（H）mm

MZ-721：440（W）×305（D）×86（H）mm

MZ-711：440（W）×305（D）×86（H）mm

重量 MZ-731：4.6 kg

MZ-721：4.0 kg

MZ-711：3.6 kg

附件：

盒式磁带（A面为BASIC，B面是应用程序）

说明书，操作表，电源电缆，连接电视机电缆。后面有彩色绘图打印机的附件表。

A．2．2

CPU插件板技术条件

CPU：LH 0080A（Z80A）……1

PPI：8255……1可编程序外围接口

PIT：8253……1

存储控制器：

〔CRTC〕：M 60719……1

ROM：监控4K字节ROM……1

字符发生器：2K字节ROM……1

RAM：64K字节D—RAM……8

2K字节S—RAM……2

I／0总线：扩展I／O（输入／输出）总线……1

打印机I／O总线……2（不能同时用）

盒式磁带机读／写终端……2

游戏杆终端……2

A．2、3　颜色绘图打印机技术条件说明

打印系统：使用有四种颜色的园珠笔

颜　色：（1）黑色（2）兰色（3）绿色（4）红色

打印速度：使用最小字符打印时，平均10个字符／秒。

行　宽：80列，40列，26列（由软件选用）

字符数：115（含ASCII字符）

分辨率：0.2mm

附　件：卷纸一筒，元珠笔（黑、兰、绿、红）纸支架（左右）两个，纸轴一支，导纸器一个。

A．2、4　数据录音机的技术说明

类型：IEC标准压缩型盒式机械（ICE国际电工委元会）

读／放系统：2面单道

速度：4.8Cm／秒±3.5%

开关控制类型：钢琴式

控制开关　PLAY，FF，REW，STOP／EJECT与及REC键和计数器恢复接扭。

数据传递方法：夏普PWM法

数据传递速度：1200字符／秒

带：普通音频盒式磁带。

A．2、5

电源技术要求

（提供彩色绘图打印机，数据录音机以及主机的电源）

输入：240／220V±10%；50／60H$_2$，20W

输出：5V

A．3　BASIC错误信息表

当操作过程中出现错误时，BASIC解释程序将显示下列格式之一的错误信息。

1）＜错误类型＞error（直接方式错误）。

2）＜错误类型＞errorin行号（运行方式错误）。

在执行直接命令时或输入程序时，如果有错，则出现格式1的错误信息。在执行程序期间发现错误，则出现格式2的错误信息。

可以显示的错误信息如下所示：

错误编号	显示信息	说明
1	Syntax error 句法错误	句法错误
2	Over floW error 溢出错误	使用的数值超出规定范围或出现溢出错误
3	Illegal data error 非法数据	使用了非法的常数和变量
5	String length error 字串太长	字串长度超过了 255 个字符
6	Memory capacity error 内存容量错	内存容量不足
7	Array def.error 重定义数组	重新定义一个比原先定义更大的数值
8	Linelength error 行长度错	行长度太长
10	GOSUB nesting error 嵌套错	GOSUB嵌套层数超出可用内存区的界限
11	FOR～NEXT error	FOR～NEXT循环的嵌套层次超出可用内存区界限
12	DEF FN nesting error自定义函数嵌套错	自定义函数嵌套次数超出界限
13	NEXT error	使用了NEXT但没有对应的FOR
14	RETURN error	使用了RETURN但没有对应的GOSUB
15	Un def.function error 未定义函数错	调用未有定义过的函数
16	Un def.line num.error 没有定义的行号	参考行号在程序中不存在
17	Can't continue 不能继续	CONT命令不能执行
18	Memory protection 内存保护	试图把数据写入BASIC的控制区
19	Instruction error 指令错	直接方式的命令和语句混在一起
20	Can't RESUME error	不能执行RESUME
21	RESUME error	没有错误出现时，企图执行RESUME命令
24	READ error	用了READ语句，却没有对应的DATA语句
43	Already open error 已经打开	文件已经打开又用了OPEN语句
63	Out of file error 无文件	读不存在的文件
65	Printer is not ready 没有准备打印机	未连接打印机
68	Printer mode error 打印机方式错	彩色绘图打印机方式错
70	Check sum error 校对总和错	校对总和错（在读带时）

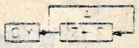
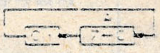
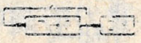
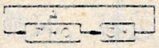
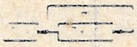
A.4 Z 80A 指令系统

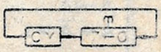
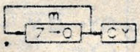
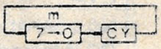
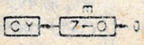
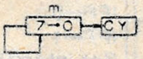
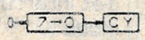
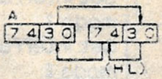
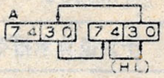
Z 80A指令简介如下，供参考

记忆符	同符号表示的操作	OP —码
八位取数指令组		
LD r, r	r←r	01 r r
LD r, n	r←n	00 r 110 ← n →
LD r, (HL)	r←(HL)	01 r 110
LD r, (IX+d)	r←(IX+d)	11 011 101 01 r 110 ← d →
LD r, (IY+d)	r←(IY+d)	11 111 101 01 r 110 ← d →
LD (HL), r	(HL)←r	01 110 r
LD (IX+d), r	(IX+d)←r	11 011 101 01 110 r ← d →
LD (IY+d), r	(IY+d)←r	11 111 101 01 110 r ← d →
LD (HL), n	(HL)←n	00 110 110 ← n →
LD (IX+d), n	(IX+d)←n	11 011 101 00 110 110 ← d → ← n →
LD (IY+d), n	(IY+d)←n	11 111 101 00 110 110 ← d → ← n →
LD A, (BC)	A←(BC)	00 001 010
LD A, (DE)	A←(DE)	00 011 010
LD A, (nn)	A←(nn)	00 111 010 ← n → ← n →
LD (BC), A	(BC)←A	00 000 010
LD (DE), A	(DE)←A	00 010 010
LD (nn), A	(nn)←A	00 110 010 ← n → ← n →
LD A, I	A←I	11 101 101 01 010 111
LD A, R	A←R	11 101 101 01 011 111
LD I, A	I←A	11 101 101 01 000 111
LD R, A	R←A	11 101 101 01 001 111
十六位取数指令组		
LD dd, nn	dd←nn	00 dd0 001 ← n → ← n →
LD IX, nn	IX←nn	11 011 101 00 100 001 ← n → ← n →
LD IY, nn	IY←nn	11 111 101 00 100 001 ← n → ← n →

记忆符	[illegible]符号表示操作	OP —码
LD HL, (nn)	H←(nn+1) L←(nn)	00 101 010 ← n → ← n →
LD dd, (nn)	dd_H←(nn+1) dd_L←(nn)	11 101 101 01 dd1 011 ← n → ← n →
LD IX, (nn)	IX_H←(nn+1) IX_L←(nn)	11 011 101 00 101 010 ← n → ← n →
LD IY, (nn)	IY_H←(nn+1) IY_L←(nn)	11 111 101 00 101 010 ← n → ← n →
LD (nn), HL	(nn+1)←H (nn)←L	00 100 010 ← n → ← n →
LD (nn), dd	(nn+1)←dd_H (nn)←dd_L	11 101 101 01 dd0 011 ← n → ← n →
LD (nn), IX	(nn+1)←IX_H (nn)←IX_L	11 011 101 00 100 010 ← n → ← n →
LD (nn), IY	(nn+1)←IY_H (nn)←IY_L	11 111 101 00 100 010 ← n → ← n →
LD SP, HL	SP←HL	11 111 001
LD SP, IX	SP←IX	11 011 101 11 111 001
LD SP, IY	SP←IY	11 111 101 11 111 001
PUSH qq	(SP−2)←qq_L (SP−1)←qq_H	11 qq0 101
PUSH IX	(SP−2)←IX_L (SP−1)←IX_H	11 011 101 11 100 101
PUSH IY	(SP−2)←IY_L (SP−1)←IY_H	11 111 101 11 100 101
POP qq	qq_H←(SP+1) qq_L←(SP)	11 qq0 001
POP IX	IX_H←(SP+1) IX_L←(SP)	11 011 101 11 100 001
POP IY	IY_H←(SP+1) IY_L←(SP)	11 111 101 11 100 001
交换指令和数据块转移和搜索指令		
EX DE, HL	DE↔HL	11 101 011
EX AF, AF'	AF↔AF'	00 001 000
EXX	(BC)↔(BC') (DE)↔(DE') (HL)↔(HL')	11 011 001
EX (SP), HL	H↔(SP+1) L↔(SP)	11 100 011
EX (SP), IX	IX_H↔(SP+1) IX_L↔(SP)	11 011 101 11 100 011
EX (SP), IY	IY_H↔(SP+1) IY_L↔(SP)	11 111 101 11 100 011

I 记忆符	用符号表示的操作	O P —码
LDI	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC−1	11 101 101 10 100 000
LDIR	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC−1 Repeat until BC=0	11 101 101 10 110 000
LDD	(DE)←(HL) DE←DE−1 HL←HL−1 BC←BC−1	11 101 101 10 101 000
LDDR	(DE)←(HL) DE←DE−1 HL←HL−1 BC←BC−1 Repeat until BC=0	11 101 101 10 111 000
CPI	A−(HL) HL←HL+1 BC←BC−1	11 101 101 10 100 001
CPIR	A−(HL) HL←HL+1 BC←BC−1 Repeat until A= (HL) or BC=0	11 101 101 10 110 001
CPD	A−(HL) HL←HL−1 BC←BC−1	11 101 101 10 101 001
CPDR	A−(HL) HL←HL−1 BC←BC−1 Repeat until A= (HL) or BC=0	11 101 001 10 111 001
八位算术指令和逻辑指令		
ADD A, r	A←A+r	10 000 r
AD A, n	A←A+n	11 000 110 ← n →
ADD A, (HL)	A←A+(HL)	10 000 110
ADD A, (IX+d)	A←A, (IX+d)	11 011 101 10 000 110 ← d →
ADD A, (IY+d)	A←A+(IY+d)	11 111 101 10 000 110 ← d →
ADC A, s	A←A+s+CY	001
SUB s	A←A−s	010
SBC A, s	A←A−s−CY	011
AND s	A=A∧s	100
OR s	A←A∨s	110
XOR s	A←A⊖s	101
CP s	A−s	111
INC r	r←r+1	00 r 100
INC (HL)	(HL)←(HL)+1	00 110 100
INC (IX+d)	(IX+d) ←(IX+d)+1	11 011 101 00 110 100 ← d →
INC (IY+d)	(IY+d) ←(IY+d)+1	11 111 101 00 110 100

I 记忆符	用符号表示的操作	O P —码
DEC m	m←m−1	← d → 101
专用算术指令和控制指令		
DAA	Decimal adjustment upon contents of A after add or subtract	00 100 111
CPL	A←$\overline{A}$	00 101 111
NEG	A←$\overline{A}$+1	11 101 101 01 000 100
CCF	CY←$\overline{CY}$	00 111 111
SCF	CY←1	00 110 111
NOP	No operation, but PC is incremented.	00 000 000
HALT	CPU halted	01 110 110
DI	IFF←0	11 110 011
EI	IFF←1	11 111 011
IM0	Set interrupt mode 0	11 101 101 01 000 110
IM1	Set interrupt mode 1	11 101 101 01 010 110
IM2	Set interrupt mode 2	11 101 101 01 011 110
十六位算术指令		
ADD HL, ss	HL←HL+ss	00 ss1 001
ADC HL, ss	HL←HL+ss+CY	11 101 101 01 ss1 010
SBC HL, ss	HL←HL−ss−CY	11 101 101 01 ss0 010
ADD IX, pp	IX←IX+pp	11 011 101 00 pp1 001
ADD IY, rr	IY←IY+rr	11 111 101 00 rr1 001
INC ss	ss←ss+1	00 ss0 011
INC IX	IX←IX+1	11 011 101 00 100 011
INC IY	IY←IY+1	11 111 101 00 100 011
DEC ss	ss←ss−1	00 ss1 011
DEC IX	IX←IX−1	11 011 101 00 101 011
DEC IY	IY←IY−1	11 111 101 00 101 011
循环和移位指令		
RLCA		00 000 111
RLA		00 010 111
RRCA		00 001 111
RRA		00 011 111
RLC r		11 001 011 00 000 r
RLC (HL)		11 001 011 00 000 110

记忆符	同符号表示操作	OP—码
RLC (IX+d)		11 011 101 11 001 011 ← d → 00 $\boxed{000}$ 110
RLC (IY+d)		11 111 011 11 001 011 ← d → 00 $\boxed{000}$ 110
RL m		$\boxed{010}$
RRC m		$\boxed{001}$
RR m		$\boxed{011}$
SLA m		$\boxed{100}$
SRA m		$\boxed{101}$
SRL m		$\boxed{111}$
RLD		11 101 101 01 101 111
RRD		11 101 101 01 100 111

位操作，复位和测试指令

BIT b, r	Z←$\overline{rb}$	11 001 011 01 b r
BIT b, (HL)	Z←$\overline{(HL)b}$	11 011 011 01 b 110
BIT b, (IX+d)	Z←$\overline{(IX+d)b}$	11 011 101 11 001 011 ← d → 01 b 110
BIT b, (IY+d)	Z←$\overline{(IY+d)b}$	11 111 101 11 001 011 ← d → 01 b 110
SET b, r	rb←1	11 001 011 $\boxed{11}$ b r
SET b, (HL)	(HL)b←1	11 001 011 $\boxed{11}$ b 110
SET b, (IX+d)	(IX+d)b←1	11 001 101 11 001 011 ← d → $\boxed{11}$ b 110
SET b, (IY+d)	(IY+d)b←1	11 111 101 11 001 011 ← d → $\boxed{11}$ b 110
RES b, m	mb←0	$\boxed{10}$

记忆符	閘符号表示操作	OP—码

转移指令

JP nn	PC←nn	11 000 011 ← n → ← n →
JP cc, nn	If condition cc is true, PC< −nn; otherwise, continue	11 cc 010 ← n → ← n →
JR e	PC←PC+e	00 011 000 ← e-2 →
JR C, e	If C=0, continue. If C=1, PC←PC+e	00 111 000 ← e-2 →
JR Z, e	If Z=0, continue If C=1, PC←PC+e	00 101 000 ← e-2 →
JR NC, e	If C=1, continue. If C=0, PC←PC+e	00 110 000 ← e-2 →
JR NZ, e	If Z=1, continue. If Z=0, PC←PC+e	00 100 000 ← e-2 →
JP (HL)	PC←HL	11 101 001
JP (IX)	PC←IX	11 011 101 11 101 001
JP (IY)	PC←IY	11 111 101 11 101 001
DJNZ e	B←B−1 If B=0, continue; otherwise, PC←PC+e	00 010 000 ← e-2 →

（调用和返回指令

CALL nn	(SP−1)←PC_H (SP−2)←PC_L PC←nn	11 001 101 ← n → ← n →
CALL cc, nn	If condition cc is false, continue; otherwise same as CALL nn.	11 cc 100 ← n → ← n →
RET	PC_L←(SP) PC_H←(SP+1)	11 001 001
RET cc	If condition cc is false, continue; otherwise same as RET.	11 cc 000
RETI	Return from interrupt	11 101 101 01 001 101
RETN	Return from $\overline{NMI}$.	11 101 101 01 000 101
RST p	(SP−1)←PC_H (SP−2)←PC_L PC_H←0 PC_L←p	11 t 111

记忆符	用符号表示的操作	OP —码
输入和输出指令		
IN A,(n)	A←(n)	11 011 011 ← n →
IN r,(C)	r←(C)	11 101 101 01 r 000
INI	(HL)←(C) B←B−1 HL←HL+1	11 101 101 10 100 010
INIR	(HL)←(C) B←B−1 HL←HL+1 Repeat until B=0	11 101 101 10 110 010
IND	(HL)←(C) B←B−1 HL←HL−1	11 101 101 10 101 010
INDR	(HL)←(C) B←B−1 HL←HL−1 Repeat until B=0	11 101 101 10 111 010

记忆符	用符号表示的操作	OP —码
OUT (n),A	(n)←A	11 010 011 ← n →
OUT (C),r	(C)←r	11 101 101 01 r 001
OUTI	(C)←(HL) B←B−1 HL←HL+1	11 101 101 10 100 011
OTIR	(C)←(HL) B←B−1 HL←HL+1 Repeat until B=0	11 101 101 10 110 011
OUTD	(C)←(HL) B←B−1 HL←HL−1	11 101 101 10 101 011
OTDR	(C)←(HL) B←B−1 HL←HL−1 Repeat until B=0	11 101 101 10 111 011

(注意) 上表符号使用方法如下

r, r'	寄存器
000	B
001	C
010	D
011	E
100	H
101	L
111	A

dd, ss	寄存器配对
00	BC
01	DE
10	HL
11	SP

qq	寄存器配对
00	BC
01	DE
10	HL
11	AF

pp	寄存器配对
00	BC
01	DE
10	IX
11	SP

rr	寄存器配对
00	BC
01	DE
10	IY
11	SP

∧ : AND 运算
∨ : OR 运算
⊕ : 异或运算

b	位操作
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

cc	条件	
000	NZ	non zero
001	Z	zero
010	NC	non carry
011	C	carry
100	PO	parity odd
101	PE	parity even
110	P	sign positive
111	M	sign negative

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

s: r, n, (HL), (IX+d), (IY+d)
CY: 进入触发器 ip
(寄存器对) H: 高八位寄存器对

m : r, (HL), (IX+d), (IY+d)
mb: b 位或条件 m
(寄存器对) L: 低八位寄存器对

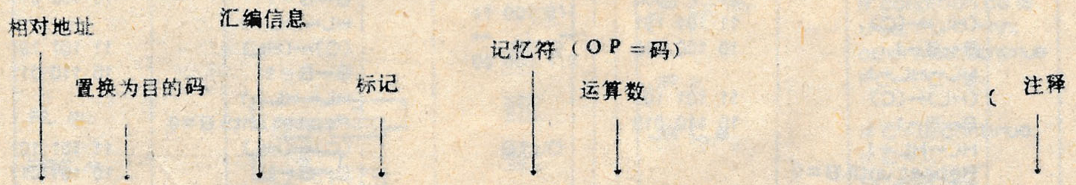
OP —码: ADC, SUB, SBC, AND, OR, XOR 和 CP. □ 空格位用 ADD 操作代替。

OP —码 DEC, □ 空格用 INC 操位代替。类似操作适用于循环，移位和位操作，复位和调试指令的 OP —码。

A.5 监控程序汇编列表

下面提供 1Z—013A监控汇编列表

汇编表由Z80汇编软件DOS产生。其符号含义如下。



```
20 02A7 13                  INC    DE
21 02A8 13                  INC    DE
22 02A9 13                  INC    DE
23 02AA C9                  RET
24 02AB              ;
25 02AB              ;
26 02AB              ;
27 02AB              ;ORG 02ABH ; MLDST
28 02AB              ;
29 02AB              ;    MELODY START & STOP
30 02AB              ;
31 02AB              MLDST:  ENT
32 02AB 2AA111               LD     HL,(RATIO)
33 02AE 7C                   LD     A,H
34 02AF B7                   OR     A
35 02B0 280C                 JR     Z,MLDSP
36 02B2 D5                   PUSH   DE
37 02B3 EB                   EX     DE,HL
38 02B4 2104E0               LD     HL,CONTO
39 02B7 73                   LD     (HL),E
40 02B8 72                   LD     (HL),D
41 02B9 3E01                 LD     A,1
42 02BB D1                   POP    DE
43 02BC 1806                 JR     MLDS1
44 02BE              ;
45 02BE              MLDSP:  ENT
46 02BE 3E36                 LD     A,36H              ; MODE SET (8253 C0)
47 02C0 3207E0               LD     (CONTF),A          ; E007H
48 02C3 AF                   XOR    A
49 02C4 3208E0       MLDS1:  LD     (SUNDG),A          ; E008H
50 02C7 C9                   RET                       ; TEHRO RESET
```

由于监控 1Z —013A 的起始地址设置为 $ 0000，在汇编表中置换地址和目的地址可分别采用绝对地址和目的码。

本汇编表仅作为参考，夏普公司对上述有关问题不承担答复的责任。

注意：本监控不同于BASIC编释系统中的监控程序。

监控程序见英文说明书第 1 6 5 页。

A.6 彩色绘图打印机的控制代码

A.6、1

在文本方式中使用的控制代码

- 文本方式（$01）
- 绘图方式（$02）……与BASIC方式的GR语句相同，置打印于绘图方式。
- 上跳一行（$03）……与BASIC的SKIP—1语句相同，把打印纸朝反向移动一行，行计数器减一。
- 检查笔（$04）……与BASIC的TEST语句相同，用四种颜色画出四个大小相同的正方形，然后置字体大小为1（40个字符／行）颜色为0（即黑色）

□ □ □ □

黑 兰 绿 红

- 缩小字符的尺寸（$09）+（$09）+（$09）

  把字符大小为1缩为0（80个字符／行）
- 取消缩小（$09）+（$09）+（$OB）

  把字符尺寸从0扩大为1，（40个字符／行）
- 置行计数器（$09）+（$09）+（ASCII）$_2$ +（ASCII）$_1$ +（ASCII）$_0$ +（$OD） 与BASIC的PAGE语句相同。

ASCII代码的三个字节指明每页规定行数，最大是255行／页，接通电源或恢复系统后，自动设置为66／页。

- 换行（$OA）……与BASIC的SKIP1语句相同。

  即打印纸向前移动一行，行的计数器加1。
- 扩大字符尺寸（$OB）

  把字符大小从1扩大为2（26个字符／行）
- 取消扩大（$OC）

  把字符大小从2缩小为1。
- 回车（$OD）

  使车架回到打印机的最左边。
- 反向空格（$OE）

  使车架向左移一个字符位置，当车架已经在打印机左边时，这个代码被忽略。
- 格式馈送（$OF）

  把打印纸移到下页的起始行，且恢复行计数器为0。
- 下一种颜色（$ID）

  把色笔依次换为下一颜色。

A.6、2

字符大小

- 接通电源时，字符大小自动置为 1（40个字符／行）以后可以通过控制代码和命令改变。
- 在绘图方式时，字符尺寸的变化范围是 0～63。
- 当绘图方式回到文本方式时，字符尺寸被置为 1。

A．6、3　绘图方式命令

在绘图方式中，由以下的输出命令控制打印机进行打印。在园括号内的词是BASIC语句，这些语句与绘图方式的命令有相同功能。

命令名	格式	功能
线型	Lp（p＝0～15）	依P值指定划线类型 P＝0为实线，P＝1～15为点虚线
打印机初始化	A	在文本模式中连通打印机
返回原点（PHOME）	H	命令笔返回原点
（HSET）	I	命令当前笔的位置作为原点（X＝0，Y＝0）
划线（LINE）	Dx, y, …xn, yn （$-999 \leqq x, y \leqq 999$）	从当前笔位置向坐标（X，Y）划线一直划到（X_2，Y_2），等等
增量划线（RLINE）	J△x，△y…△xn，△yn （$-999 \leqq \triangle x, \triangle y \leqq 999$）	从当前笔位置依增量（△X，△Y）划线一直划到（$\triangle X_2$，$\triangle Y_2$）等等
抬笔指令（MOVE）	Mx，y （$-999 \leqq x, y \leqq 999$）	抬笔移至坐标（X，Y）处
增量移笔指令（RMOVE）	R△x，△y （$-999 \leqq \triangle x, \triangle y \leqq 999$）	笔左移量为（△X，△Y）增量值
换笔指令（PCOLOR）	Cn（n＝0～3）	依给定n值进行换笔
设置字符大小	Sn（n＝0～63）	依n值指令字符尺寸
打印转向指令	Qn（n＝0～3）	依n值指定字符打印方向
打印指令	P$c_1c_2c_3\cdots c_n$（n＝∞）	打印字符
划坐标轴指令（AXIS）	Xp，p，r（p＝0 or 1） （p＝－999～999） （r＝1～255）	当P＝1时划X轴 当P＝0时划Y轴 q为X或Y最大长度 r为标尺个数

当打印机从绘图方式转换为文本方式时，将改变如下数据。

- X和Y座标

  Y座标置0，且原点在可打印区的左边。
- 打印字符方向

  Q被置0
- 字符尺寸

  字符大小置为1
- 线的型式不变。

A．7、注意有关操作

■数据录音机

- 虽然MZ—700机的数据录音机是最可仗赖的，但在使用一段时间之后，读/写头总是会磨损的。此外，磁性颗粒和灰尘可能会积存在磁头，读/写功能也会退化，因此必须定期地清洗磁头。当磁头已经磨损时，就要更换。

1）为了清洁磁头，要打开盒箱，按下 PLAY 键擦净磁头和用棉花擦洗压轮。如果它们很脏，应用浸渍过洒精的棉花擦洗。

2）当磁头磨损时，要与经营者联系，不要试图自己更换磁头。

■盒式磁带

- MZ — 700机可以使用任何商业上通用的盒式磁带，但还是建议你使用可信赖的厂家生产的品质优良的盒式磁带为好。
- 使用普通型号的磁带
- 避免用C—120型的盒式磁带。
- 建议使用C—60或更短的盒式磁带。
- 在装入盒式磁带之前，如果磁带过于松弛，务必用铅笔或类似的东西把松弛的磁带卷上，否则可能会拆断磁带或会缠住压轮而损坏。
- 保护磁带中录好的程序和数据，防止意外擦掉。

MZ—700机的数据录音机，装备了保护功能。其操作方法同使用普通的音频盒式磁带机。

为了防止意外地擦掉磁带上的数据，应用螺丝刀或类似的东西，把录了数据的磁带的盒子上的锁片挑开，这就使得 RECORD 键不可能被按下，从而防止了擦掉有意义的数据。

■其它

- 关于商业上通用的盒式磁带机参见第三章3·3
- 显示器

当使用的显示器不是MZ—700专用显示器时，屏幕大小必须调整。

■彩色绘图打印机

- 换笔时，在倒转方向上笔的鼓轮不旋转。
- 如果在一段时间内不使用打字机时，为了防止笔水干涸，务必从鼓轮上把笔取下。

然后盖上笔帽，放到笔盒内保存起来。

- 打印的字符，要等一定的时间才能干（墨水是一种水溶液）
- 打印机的盖移开后，不要撕打印纸，撕打印纸时，要抓住纸架上的纸。
- 换笔时，不要摸内部机械，如果不足以注意这个提议，结果很可能会损坏打印机。
- 接通电源时，彩色绘图打印机会产生一些声音，这是正常现象。
- 一行打印80个字符，可能不便阅读，在这种情况下，可使用每行打印40个字符
- 在绘图方式时，重复打印的线条可能会变得模糊，重复打印虚线时，特别容易出现这种现象，这是园珠笔固有的特点，难于避免。

■注意有关软件

- 载入BASIC解释程序时，大约要花3分钟。
- 在如下情况中使用计算机后面的复位开关（参见3、1、1）

在正常执行期间或者程序进入死循环时，用以停止BASIC程序执行，为了返回程序，使用#命令，然而，如果程序循环不已，应检查程序或硬件。

■BASIC的计算误差

- 在执行计算之前，BASIC要把十进制数转换成浮点二进制数。为了显示又把二进制的计算结果转换成十进制数，这就可能使计算结果产生一定误差。

例：PRINT　817.3—810.4

6.899999……正确结果是6.9

- 在计算函数和指数时，结果是近似值。
- 在使用IF语句时，上面的情形必须考虑。

例如：

```
 10 A＝1／100＊100
 20 IF A＝1  THEN  PRINT  “TRUE”；GOTO  40
 30 PRINT  “FALSE”
 40 PRINT “A＝”；A
 50 END
RUN
FALSE
A＝1
```

虽然在10行方程式的结果是1，但这个程序打印出FALSE，是因为转换时出现误差所至。

■注意有关处理

- 电源开关

电源开关在拨向通或关位置后，最少在10秒钟内不要拨动，这一点对于保证正确地操作计算机是必要的，当电源开关在通的位置时，不要拔去电源电缆，否则可能会招来事故。

- 电源电缆

避免把重的物体如配电盘压着电源电缆，否则可能会损坏它，甚至会产生严重事故。在拔出它时，务必要抓住电缆的插塞拔。

- 供电电压

供电电压是 240／220 VAC（交流电压）如果电压过高或过低，计算机就不能正常地工作。如果你遇到这个问题应与你的购买商联系，以求帮助解决。

- 通风

在计算机的电源箱上，提供了许多通风孔，以防止过热，使用时，计算机应放在通风处。且不要用布盖住，计算机的左边不要放任何东西，因为这是电源部件的通风所在。

- 潮湿和灰尘

计算机不要放在潮湿或有灰尘的地方。

- 环境条件

不要把计算机放在热源附近或太阳直接照射的地方。如果不注意这一点，可能会损坏计算机的元件。

- 水和外来物质

水和其它外来物质（如锁碎物之类）落入计算机内，将会损坏计算机。如果出现意外事件，在拔出电源，电缆后，应与你购买计算机的商人联系，以求帮助解决。

- 震动

避免计算机遭受震动，强烈震动，可能会造成计算机永久性损坏。

- 事故

你如果发现任何不正常现象，应马上停止操作，并与计算机商联系。

- 暂时不用

如果你的计算机暂时不用时，应拔去电源电缆。

- 关于周边设备

关于周边设备，你只能使用夏普公司设计的部件和元件，否则可能会损坏计算机。

- 尘

当计算机箱脏了时，应用浸过水或洗涤剂的软布擦净，以避免计算机箱褪色。不要用挥发的液体如笨类擦。

- 噪声

当供给计算机的交流电源，存在音电平的噪声信号时，建议你使用一个线路滤波器（可以从夏普商人那里购买）同时应把信号电缆，尽可能地远离电源电缆或其它电器设备。

- 视频干扰

如果在计算机的附近使用电视机或接收机的话，可能会出现干扰，这是因为计算机产生的视频信号的原因。这样电视机可能会产生一个强的磁场，使计算机不能正确地工作。在这种情况下，电视机最少要远离计算机 2～3米。

MZ—700 的 AASIC 解释程序的复制修改

A：下面叙述BASIC带的复制过程

1）接通MZ－731电源，屏幕显示"＊＊MONITOR 1Z－013A＊＊"表示计算机进入monitor state（监控状态）＊号表示监控状态下等待命令.

2）用监控命令M修改部分内存，过程如下：

在＊号提示后键入"MOF 00 ↵"随后屏幕显示和键入内容如下：

屏幕显示	键入
＊MCF 00	
CF 00FF	CD ↵
CF 01FF	27 ↵
CF 02FF	00 ↵
CF 03FF	38 ↵
CF 04FF	E3 ↵
CF 05FF	CD ↵
CF 06FF	2A ↵
CF 07FF	00 ↵
CF 08FF	DA ↵
CF 09FF	FE ↵
CF 0AFF	00 ↵
CF 0BFF	E3 ↵
CF 0CFF	AD ↵
CF 0DFF	00 ↵
CF 0EFF	CD ↵
CF 0FFF	27 ↵
CF 10FF	00 ↵
CF 11FF	38 ↵
CF 12FF	F5 ↵
CF 13FF	C3 ↵
CF 14FF	CB ↵
CF 15FF	OF ↵

[SHIFT] 与 [BREAK] 键一块按下，屏幕显示＊号

＊ ▓ —光标

3）把BASIC的原带装入数据录音机

4）键入监控命令J如下：

＊ JCF 00 ↵　　显示

↓ PLAY

注意如果磁带录音机上已有某一按钮处于按下状态，则没有此提示。

5）根据提示符“↓ PLAY”，按下 PLAY 按钮，开始载入BASIC内容，在这种情况下没有文件名（FILE NAME）等提示，此时如果出现错误就要重新从第（1）项起重做。正常载入结束时出现星号（*），然后把BASIC原带绕回取出，把一盒新空白带装在数据录音机上。

6）键入如下

```
* J 1108↵
```

7）此时清屏，然后显示如下：

```
S—BASIC  EX  SAVER  丰 16 ⁙
HIT  ANY  KEY？
```

8）按下任何一个键屏幕进一步显示

```
↓ RECORD  PLAY
```

9）按下 RECORD 按钮，开始复制，屏幕将显示

```
WRITING  S—BASIC
```

10）复制完成后，计算机响“plt plt”两声，同时清屏并再次显示S—BASIC EX SAVER 05丰 16 ⁙

```
HIT  ANY  KEY？⁙
```

此时把复了的磁带绕回，如果还要复制，则取出磁带盒，然后装入新的空白带，从第8项起重做，需要复制几份就这样重做几次。

11）不需再复制的话，请按一下计算机背后的RESET开关，计算机则返回监控状态，此时屏幕显示。

```
** MONITOR  1Z—013A **
* ⁙
```

12）把磁带反绕到复制前的位置然后按下STOP按钮。

13）在*号提示下键入如下

```
* JCFOE↵
↓ PLAY
```

14）此时按下录音机上的 PLAY 按钮，计算机将开始执行校对，当校对无误时，将显示“OK！”但没有FILE NAME等显示。如果出现错误，应从第4项起重新做。

15）录好的磁带务必卸掉防抹片，以实现保护磁带录制内容。

B：修正BASIC解释程序内容的步骤如下：

a）从第1到第5项的操作与上面复制时的情形完全相同。

b）利用监控程序命令M修改访问地址。

实例：把地址1234H中的8A改为7A

		键入
*M		1234
1234	8A	7ACR

附　录A 8

A 8 ·1 如何编写程序

第一节　编写程序的步骤

如何编写解决实际计算问题的程序呢？下面通过实例介绍编写程序的大致步骤。例如：有任意排列的一组数，现要求把这些数加以整理，使之按从大到小顺序排列，试编写程序。

编写程序的步骤如下：

A8·1、1　确定计算方案和步骤

分析：因为要将这几个数从大到小顺序排列，必须反复使用到这几个数据，故此应先把这几个数读入到一个一维数组A里，如果按照题意整理了数组A里的下标变量，那么，在这个数组里，下标变量A（1）的值应该是所有下标量中最大的，A（2）的值较A（1）小，而A（3）的值较A（2）小，依此类推，A（n）的值是最小的。

计算步骤如下：

先把A（1）的值与其它所有下标变量（2）—A（n）的值一一比较，如A（1）＜A（2），就将A（2）的值赋于 A（1），原A（1）的值赋于 A（2）．如A（1）≥A（2），则A（1）原值不动，再和A（3）比，如A（1）≥A（3），则

继续和A（4）比，如A（1）＜A（3），则将A（3）赋予A（1），A（1）的值赋予A（3），这样一直比到A（n），这时A（1）的值就是最大的了。

再求次大值。将A（2）的值用上述同样方法和后面所有下标变量（A（3）～A（n））的值一一比较最后所得的A（2）的值就是n个数中的次大值。

按同样方法一直做下去，一直做到A（n－1）和A（n）相比后为止，这时剩下的A（n）就是整个数组中的最小值。

比较过程示意如下：

第一次：　A（1）⟷A（2）
　　　　　A（1）⟷A（3）
　　　　　A（1）⟷A（n—1）
　　　　　A（1）⟷A（n）
第二次：　A（2）⟷A（3）
　　　　　A（2）⟷A（4）
　　　　　A（2）⟷A（n—1）
　　　　　A（2）⟷A（n）
第n—1次：A（n—1）⟷A（n）

A8·1、2

根据计算步骤，画出程序框图

框图是解题步骤的图示。它不但有助于编写程序，而且有利于检查程序。

对于简单的计算问题，这一步可以省略。

关于框图的说明

1）框图中的每一框表示一个程序段（包括一个或多个语句）的功能。

2）框分为二类：一类是矩形框，表示要进行的工作，它通常有一个入口和一个出口。另一类是菱形框，它表示要进行比较和判断，并按一定条件选取一条路径；它有一个入口和二个出口，即比较后，形成两个分支，在两个出口处必须注明那一条分支是对应于满足条件的（常以“是”表示），那--条分支是对应于不满足条件的（以“否”表示）。

3）箭头表示执行顺序的流向。

在这个实例中，如果用I表示用来比较的下标变量的下标，则I从1到n—1；如果用J表示被比较的下标变量的下标，则对于每个I，J都可从I＋1变化到n。画出的程序框图见189页。

A8·1、3

确定程序结构

在这个实例中可采用嵌套的循环语句。J. I分别为内外循环变量，内循环的任务是使第一个下标变量的数值最大，外循环的任务是使第 J 个下变量的数值为第 I 大。

A8·1、4

确定其它有关问题

一个BASIC程序大体可分成三个部分：提供数据，进行运算，输出结果。以上确定了运算步骤，以下再确定有关问题。

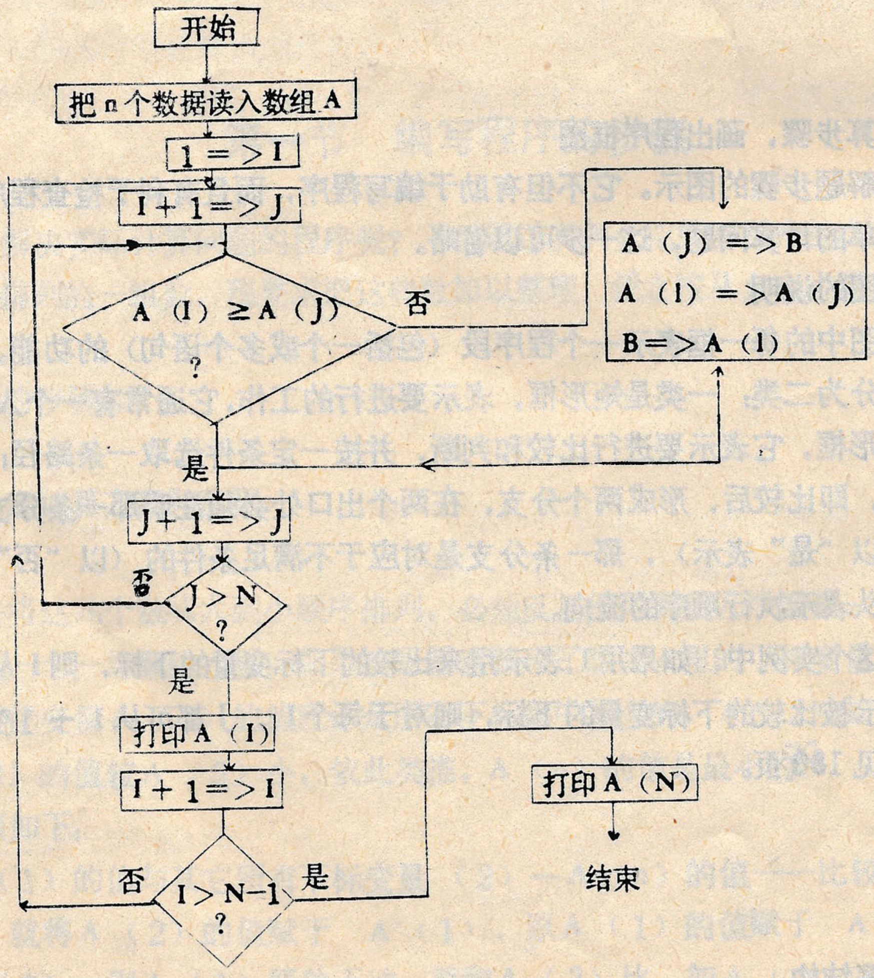
1）原始数据输入形式

为了使输入数据具有灵活性，可采用键盘输入语句。

2）安排好计算结果的输出形式

从程序框图里可以看到打印语句放于每次内循环结束之处，这是因为只有经过内循环运算后所得到的A（I）才是下标变量A（I）到A（N）中值最大的。由于I的终值是n—1，故此在内循环无法把A（n）打印出来，解决办法是在程序结束前再次使用打印语句。

为了方便阅读采用字符“，”来分隔计算结果各数。


开始
把n个数据读入数组A
1=>I
I+1=>J
A（I）≥A（J）?
否
A（J）=>B
A（I）=>A（J）
B=>A（I）
是
J+1=>J
否
J>N?
是
打印A（I）
I+1=>I
否
I>N−1?
是
打印A（N）
结束

A 8·1、5

按照BASIC的语法法则，写出实现计算问题的程序。

实现n个数从大到小排列的程序如下：

```
 10 INPUT "VALUE  OF  N=" ; N
 20 DIMA〔N〕
 30 PRINT "ENTER  YOUR  NUMBERS"
 40 FOR  I=1  TO N
 50 PRINTI; ":" ;
 60 INPUTA〔I〕
 70 NEXT
 80 PRINT/ P "THE  ANSWER  IS"
 90 FOR I=1     TO  N-1
100 FOR J=I+1  TO  N
110 IFA〔I〕>=A〔J〕 THEN  130
120 B=A〔J〕 : A〔J〕 =A〔I〕 : A 〔I 〕=B
130 NEXT  J
140 PRINT/ PA 〔I〕; "." ;
150 NEXT  I:PRINT/ PA  〔N〕
160 END
```

第二节　程序优化

除了能编出符合BASIC语法规则的程序之外，还要考虑程序优化问题，这里的优化指的是提高计算效率，节省计算时间和内存空间的问题，关于这方面提出以下建议，供大家参考。

1、简化程序，把不必要的行或语句消去。但要注意这并不意味着程序中的语句越少越好。

请看下例：

计算$Y=\sqrt{2}+\sqrt{X}$（当X=1，2，3……166）的值。

程序 1.

```
10 FOR  X=1  TO  100
20 PRINT  SQR（2）+SQR（X）
30 NEXT
```

程序2.

```
10 A＝SQR（2）
20 FOR X＝1 TO 100
30 PRINT A＋SQR（X）
40 NEXT
```

乍看起来程序1比程序2简单，因为程序2多用了一行，但机器在执行这两个程序时速度刚好相反，即执行程序2的时间比程序1少，因为程序2对$\sqrt{2}$只计算一次，而程序1却要计算100次，每逢执行循环体，就要计算一次$\sqrt{2}$，这就浪费了机器的时间。

影响优化程序的因素是相当复杂的，因此必须全面考虑，从中找出较优方案。

2）上述例子也表明，在编写循环语句时，要将循环体内的不变运算外提。

3）尽量压缩注释语句（REM），这样可以节省内存空间。

4）在语句、运算符以及字母、数字间省去不必要空格。因为空格既占用计算机空间，又花费计算机时间。

5）在可能情况下，使用多语句的程序行。因为每打入一个新的行号要占用5个字节

6）准确地为数组定维和分配字符串空间，这样可以充分利用有限的存贮空间。

7）在编写表达式时，尽量少用园括号，因为计算在处理括号时，必须花费较多的字节和时间。

8）计算多项式的值，尽量采用下面的第二种形式，避免使用第一种形式。

第一种形式：

```
Y＝X↑4＋2＊X↑3＋3＊X↑2＋5＊X＋1
```

第二种形式：

```
Y＝（（（X＋2）＊X＋3）＊X＋5）＊X＋1
```

因为在第一种形式里有三次调用乘幂子程序，故此执行时间长。

对于求A的B次幂A↑B，当指数是整数时，最好用连乘来代替乘方，如X^3写成X＊X＊X，而不写成X↑3，这样，可以缩短计算机的执行时间。

9）将计算中反复使用的程序段抽出来，写成子程序形式。

上面介绍的事项看起来都是很琐碎的，对大多数简单的程序来说，考虑与否无关大局，但在解决一些复杂的计算中，编写的程序很长时，如果能很好地考虑上述那些事项，也许能节省不少机时，并使有限的内存空间尽可能地胜任较大型的计算。

A8·2 几种简单计算方法的BASIC程序

几种简单计算方法的BASIC程序：

计算方法是解决近似计算的一门专门学科，它为任何复什的数学方程提供了数值计算的近似方法。例如，解代数方程的牛顿法，求数值积分辛普生公式等。而要编写数值计

算的程序除了必须具备计算机语言的程序设计知识之外，还必须懂得算题的计算方法。

第一节　用牛顿法解代数方程

用牛顿法求方程 $9-6X+4X^2-2x^3=0$ 的一个近似根，精确到0.00001，初值 $X_0=1.5$

对已知方程有

$$f(X)=9-6X+4X^2-2X^3=9-((2X-4)X+6)X$$

$$f'(X)=-6+8X-6X^2=-6-(6X-8)X$$

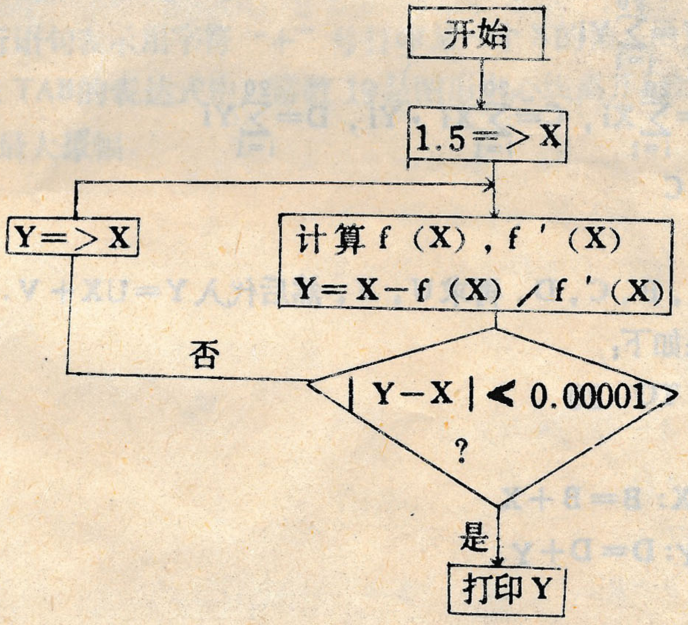
牛顿法的迭代公式为

$$Xn=Xn-1-f(Xn-1)/f'(Xn-1)\quad n=1,2\cdots\cdots$$

只要相邻两次结果之差的绝对值

$|Xn-Xn_{-1}|\leq 0.00001$，则Xn可做为方程的近似根。

解此题的程序方框图如下所示：



程序及执行的结果如下：

```
10 REM  TO  SOLVE  ALGEBRAIC  EQUATION
   BYNEW  TONIS  METHOD
   （注：用牛顿法解代数方程）
20 X＝1.5
30 A＝9－((2*X－4) *X＋6) *X
40 B＝－6－(6*X－8) *X
50 Y＝X－A/B
60 IF ABS (Y－X) <＝1.0E－5  GOTO  80
70 X＝Y: GOTO  30
```

```
80 PRINT “REAL ROOT: X=”; Y
90 END
```

第二节　配直线回归方程

实验得出下列一组对应数据

X	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Y	3.375	3.770	4.115	4.360	4.605	5.650	5.495	5.540	6.185	6.450
X	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
Y	6.775	7.220	7.865	7.210	8.055	8.530	8.445	9.290	9.535	9.880

试根据这些数据配直线回归方程

令此直线回归方程为Y＝UX＋V

根据最小二乘法有：

$$\left(\sum_{i=1}^{20} Xi^2\right) U + \left(\sum_{i=1}^{20} Xi\right) V = \sum_{i=1}^{20} Xi \cdot Yi$$

$$\left(\sum_{i=1}^{20} Xi\right) U + 20V = \sum_{i=1}^{20} Yi$$

令 $A=\sum_{i=1}^{20} Xi^2$, $B=\sum_{i=1}^{20} Xi$, $C=\sum_{i=1}^{20} Xi \cdot Yi$, $D=\sum_{i=1}^{20} Yi$

则有：　$A \cdot U + B \cdot V = C$

$BU + 20V = D$

解题的步骤：先求A，B，C，D，再求U，V，然后代入Y＝UX＋V.

计算的程序及执行的结果如下：

```
10 FOR I=1 TO 20
20 READ X,Y
30 A=A+X*X: B=B+X
40 C=C+X*Y: D=D+Y
50 NEXT
60 G=20*A-B*B
70 IF G=0 THEN PRINT/ P “NO UNIQUE SOLUIO N”: END
80 U= (20*C-B*D)/G: V=(A*D-B*C)/G
90 PRINT/ P “Y=” U; “*X+”; V
100 DATAO, 1, 3.375,  0.2, 3.770, 0.3, 4.115, 0.4, 4.360,
    0.5, 4.605, 0.6, 5.650, 0.7, 5.495, 0.8, 5.540, 0.9,
    6.185, 1.0, 6.450,
110 DATA 1.1, 6.775, 1.2, 7.220, 1.3, 7.865, 1.4, 7.210,
    1.5, 8.055, 1.6, 8.530, 1.7, 8.445, 1.8, 9.290, 1.9,
    9.535, 2.0, 9.880
```

```
RUN ↲
r= 3.3399248 * X+ 3.1105789
```

第三节　打印衰减震荡函数$e^{-0.12x}$. SINX的曲线

```
10 MODE  TS
20 PAGE  70
30 PRINT/ PTAB(0) "0" : TAB ( 19) "19" : TAB〔 38 〕 "38"
40 PRINT/ PTAB〔0〕"↓" : TAB〔 19 〕 "↓" : TAB〔 38 〕 "↓"
50 FOR X=0  TO  6*PAI (1) STEP PAI (1) / 8
60 PRINT/ PTAB〔 19+ 19*EXP〔 -.12*X 〕*SIN〔X〕 〕 "+"
70 NEXT
80 PRINT/ PTAB〔0〕"↑" : TA B〔 19〕"↑" : TAB〔 38〕"↑"
90 PRINT/ PTAB〔0〕"0" : TAB〔 19〕"19" : TAB〔 38〕"38"
```

程序中60行语句表示用字符"+"号打印某一个X的值，函数$e^{-0.12x}$. SINX取整后的近似值，在TAB的表达式中，常数19是图形中心线离开起始位置的间格数，常数19用来控制曲线的最大振幅。

第四节　用辛普生公式求数值积分

求$S=\int_0^1\sqrt{1-X^2}\,dx$之值（误差不超过E= 0.00001）

计算方法：辛普生公式：

$$\int_a^b f(X)\,dx=\frac{b-a}{6n}(y_0+4(y_1+y_3+y_5+\cdots\cdots\ y_{2n-1})+2(y_2+y_4+\cdots\cdots y_{2n-2})+y_{2n})$$

，

令$H=\frac{b-a}{2n}=\frac{b-a}{K}$为步长，其中K是划分区间（a、b）的等分数，则上述公式可改写成：

$$\int_a^b f(X)\,dx=\frac{H}{3}\left(y_0+4(y_1+y_3+\cdots\cdots+y_{2n-1})+2(y_2+y_4+\cdots\cdots+y_{2n-2})+y_{2n}\right.$$

可以通过逐次加密分点的办法，来达到要求的精确度，当加密分点前后的结果之差小于或等于E时，则认为已满足精度要求。

当采用加倍的逐次加密分点的方法时，第i次加密后，得：

$$P=Y_0+2(Y_1+Y_2+\cdots\cdots+Y_{2n-1})+Y_{2n}$$

当i＋1次加密后，各奇数点的Y值之和为D，注意到后来第i次各内节点(即除Y_0及Y_{2n}外各节点）上的Y值就是第i＋1次加密后的偶数点上的Y值，则此时积分为：

$$S=(P+4*D)*H/3$$

编写的程序及执行结果如下：

```
  10 READ A、B、E、SO，K
  20 DATA  0，1，1.0 E－5，0，1
  30 H＝B－A
  40 X＝A：GOSUB  160：P＝Y：X＝B：GOSUB  160：P＝P＋Y
  50 S＝H＊P／3
  60 IF ABS〔S－SO〕<＝E THEN  150
  70 X＝A＋H／2：SO ＝S：K＝2＊K：D＝0
  80 FOR  I＝1  TO  K－1  SIEP  2
  90 GOSUB  160：D＝D＋Y：X＝X＋H
 100 NEXT
 110 H＝H／2
 120 S＝〔P＋4＊D 〕＊H／3
 130 P＝P＋2＊D
 140 GOTO  60
 150 PRINT／P "K＝"；K，"S＝" S：END
 160 Y＝SQR〔1－X＊X〕
 170 RETURN
     RUN
 K＝ 1024，  S＝ 0.785395
```

上述计算结果表明，当把积分区间〔a，b〕划分为1024等分时，近似积分值为0.785395，

第五节　用高斯消去法解线性方　组

设含有几个未知数的线性方程组：

$$a_{11}X_1+a_{12}X_2+\cdots\cdots+a_{1n}X_n=a_{1n+1}$$

$$a_{21}X_1+a_{22}X_2+\cdots\cdots+a_{2n}X_n=a_{2n+1}$$

$$\vdots$$

$$a_{i1}X_1+a_{i2}X+\cdots\cdots+a_{in}X_n=a_{in+1}$$

$$\vdots$$

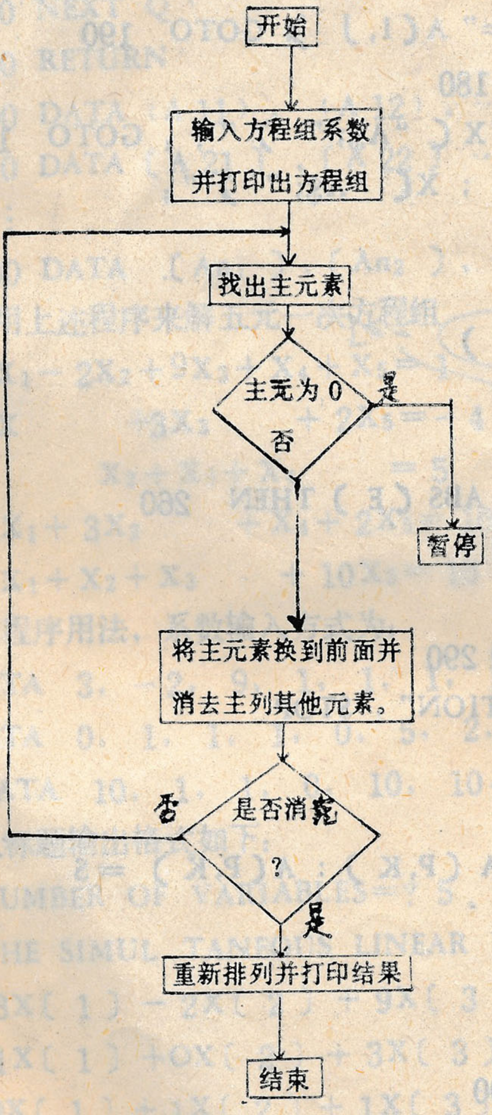
$$a_{n1}X_1+a_{n2}X_2+\cdots\cdots+a_{nn}X_n=a_{nn+1}$$

$$\sum_{i=1}^{n}AijXj=Aij+1\quad i=1,\ 2\cdots\cdots n$$

其中Aij为非异实矩阵。

求解方法：先在系数矩阵中找出最大模元素作为主元素，将其所在的行与第一行交换，并将其所在的列的其他元素消去。继而从第二行开始的n－1阶矩阵中重复上述过程，直至n行，最后将解的次序适当按排，并印出结果。若主元素为0则无单值解，下面画出框图　，和列出通用程序。可以单独使用。也可以用作子程序。用作子程序时，主程序用GOSUB 10，并将70行语句改为RETURN。程序中使用的STR$和MID$是字符串函数。

解线性方程组的框图


开始
输入方程组系数并打印出方程组
找出主元素
主元为0
是
否
暂停
将主元素换到前面并消去主列其他元素。
是否消完？
否
是
重新排列并打印结果
结束

编写的程序如下：

```
10 REMTO SQLVE SIMULTANEOUS LINEAR EQUATION
     BY GAUSSAN ELIMINTION METHOD
20 INPUT "NUMBER OF VARIABLES=" ; N
30 DIM A (N, N+1) , S (N) , M (N)
40 GOSUB 100
50 GOSUB 200
60 GOSUB 500
70 END
100 PRINT/P "THE SIMUL TANEOUS LINEAR EQUATION IS:"
110 FOR I=1 TO N
120 FOR J=1 TO N+1
130 READ A(I, J) : A$=MID$ (STR$ (J) , 1, 2)
140 IF J=1PRINT/P A(I, J) ; "X( "A$; " )" ; : GOTO 190
150 IF J=N+1 PRINT/P "=" A(I, J ): GOTO 190
160 IF A(I, J )>=O THEN 180
170 PRINT/P A(I, J ) ; "X( "A$; " ) " ; : GOTO 190
180 PRINT/P "+" A(I, J ) ; X( "A$; " ) " ;
190 NEXT J, I : RETURN
200 FOR I=1 TO N
210 P=1: Q=1: E=A (I, J ) 225
220 FOR J=1 TO N
230 FOR K=1 TO N
240 IF ABS(A (J,K) ) <=ABS (E ) THEN 260
250 E=A( J,K ) : Q=K: P=J
260 NEXT K, J
270 IF ABS(E)>1E-30 GOTO 290
280 PRINT "NO UNIOUE SOLUTION" : STOP
290 IF P=I THEN 330
300 FOR K=1 TO N+1
310 S=A(I,K ): A(I,K ) =A (P,K ) : A(P,K ) =S
320 NEXT K
330 FOR J=1 TO N
340 IF J=I GOTO 400
350 IF A( J,Q )=0 GOTO 400
360 R=A ( J·Q )/A ( I,Q )
370 FOR K=1 TO N+1
```

```
380 A[J,K]=A[J,K]-A[I,K]*R
390 NEXT K
400 NEXT J
410 M (I) =Q
420 NEXT I
430 RETURN
500 FOR I=1 TO N
510 Q=M (I) : S (Q) =A (I,N+1) / A (I,Q)
520 NEXT I
530 PRINT/ P "THE SOLVTION OF EQUA-TIONS"
540 FOR Q=1 TO N
550 B$=MID$ (STR$ (Q) ,1,2)
560 PRINT/ P "X [ "B$;" ] ="S (Q)
570 NEXT Q
580 RETURN
600 DATA (A 11) , (A 12) , ......[A1n] , [A1n+1]
610 DATA [A 21] , [A 22] ...... [A2n] , [A2n+1]
  ⋮
800 DATA .[An1] , [An2] , ...... [Ann], [Ann+1]
```

现在用上述程序来解五元一次方程组

$$\begin{cases} 3X_1-2X_2+9X_3+X_4+X_5=1 \\ X \quad +3X_3 \quad +2X_5=-4 \\ X_2+X_3+X_4 \quad =5 \\ 2X_1+3X_2 \quad +X_4+2X_5=10 \\ 10X_1+X_2+X_3 \quad +10X_5=10 \end{cases}$$

按照程序用法，系数输入方式为:

```
600 DATA 3, -2, 9, 1, 1, 1, 1, 0, 3, 0, 2, -4
610 DATA 0, 1, 1, 1, 0, 5, 2, 3, 0, 1, 2, 10
620 DATA 10, 1, 1, 0, 10, 10.
```

上机标题输出格式如下:

```
NUMBER OF VARIABLES=? 5
THE SIMUL TANEOUS LINEAR EQUATION IS:
3X[ 1 ] - 2X[ 2 ] + 9X[ 3 ] + 1X[ 4 ] + 1X[ 5 ] = 1
1X[ 1 ] +0X[ 2 ] + 3X[ 3 ] +0X[ 4 ] + 2X[ 5 ]  =-4
0X[ 1 ] + 1X[ 2 ] + 1X[ 3 ] + 1X[ 4 ] + 0X[ 5 ] = 5
2X[ 1 ] + 3X[ 2 ] +0X[ 3 ] + 1X[ 4 ] + 2X[ 5 ] = 10
```

```
10X〔 1 〕+ 1X〔 2 〕+ 1X〔 3 〕+OX〔 4 〕+ 10X〔 5 〕= 10
THE SOLUTION OF EQUA TIONS
X〔 1 〕= 3
X〔 2 〕= 1
X〔 3 〕=－1
X〔 4 〕= 5
```